

# IEEM535300

## 智慧化企業整合

# MYSQL

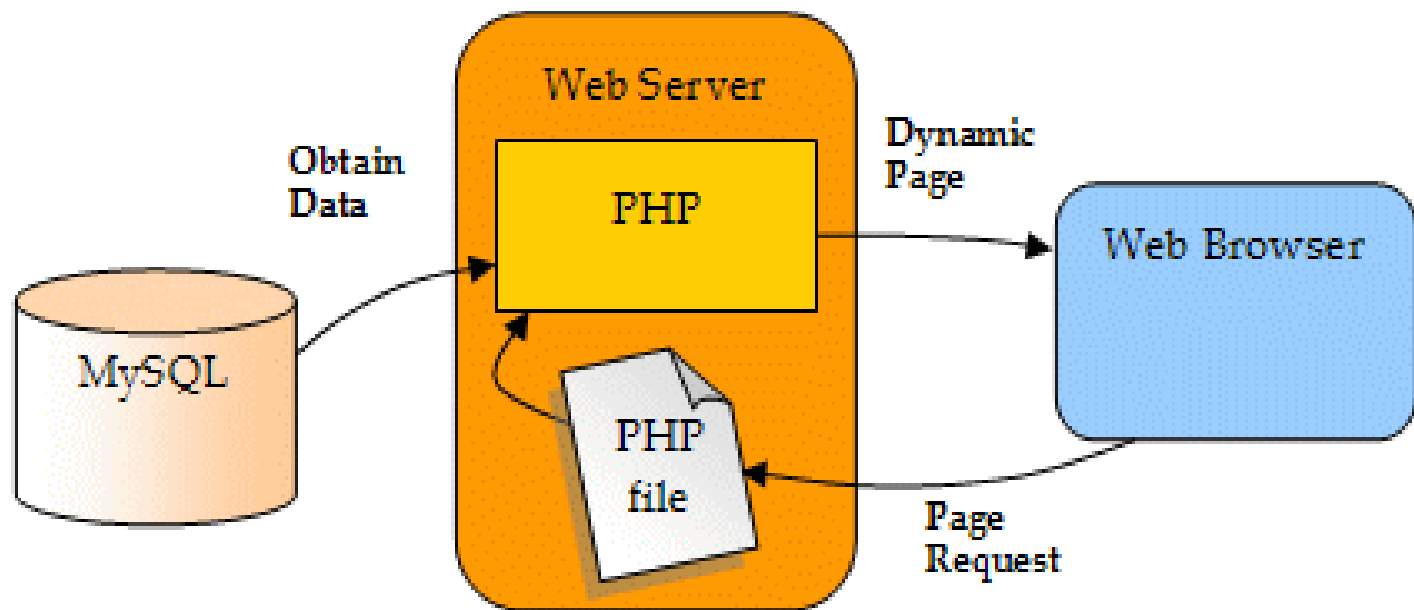
助教 陳道明

# 大綱

- **MYSQL與PHP之間的運行**
  - 創建資料庫
  - 創建資料表
  - 輸入值
  - \$ \_GET指令與\$ \_POST指令
- **PHP & MySQL物件導向應用**
  - 設定連結DB的共用常數
  - 查詢作業
  - 新增作業
  - 更新作業
  - 刪除作業
- **MYSQL實際操作**
  - 建立資料庫
  - 建立資料表
  - 主鍵設定
  - SQL語法
  - 表內插入值
  - 刪除資料表
  - 資料庫備份
  - 導入資料表

# MYSQL與PHP的運行

- MYSQL在webserver裡的作用是作為數據庫(database)。
- PHP是用來攫取數據庫裡的數據。



# Mysql 和 Mysqli的差別

- 執行速度：`mysql_query` > `mysqli_query`
- `mysqli`的用處與優點：
  - ✓ `mysqli_query`有Escape的功能
  - ✓ 直接用`mysqli_query`來處理insert to 、update set 資料庫的資料
  - ✓ `mysqli`可以防止SQL injection的攻擊
  - ✓ `mysqli`支援 SSH 加密連線

# 用 PHP 在 MYSQL 創建資料庫

- Creating Database

php: **mysqli\_connect** (主機名稱, 使用者, 密碼)

sql: **CREATE DATABASE** 資料庫名稱

```
<?php
    $con=mysqli_connect("example.com", "username", "password");
    $sql="CREATE DATABASE my_db";
    if (mysqli_query($con,$sql)) {
        echo "Database my_db created successfully";
    }
?>
```

主機位置      使用者      密碼

資料庫名稱

顯示

# 用 PHP 在 MYSQL 創建資料表

- Creating Tables

php: **mysqli\_connect** (主機名稱, 使用者, 密碼, 資料庫名稱)

sql: **CREATE TABLE** 資料表名稱 (資料表欄位)

```
<?php
$con=mysqli_connect("example.com","username","password","my_db");
$sql="CREATE TABLE table1(Username CHAR(30),Password CHAR(30),Role CHAR(30))";
if (mysqli_query($con,$sql)) {
    echo "Table have been created successfully";
}
?>
```

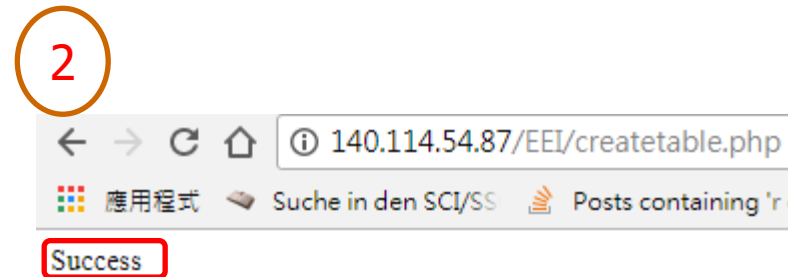
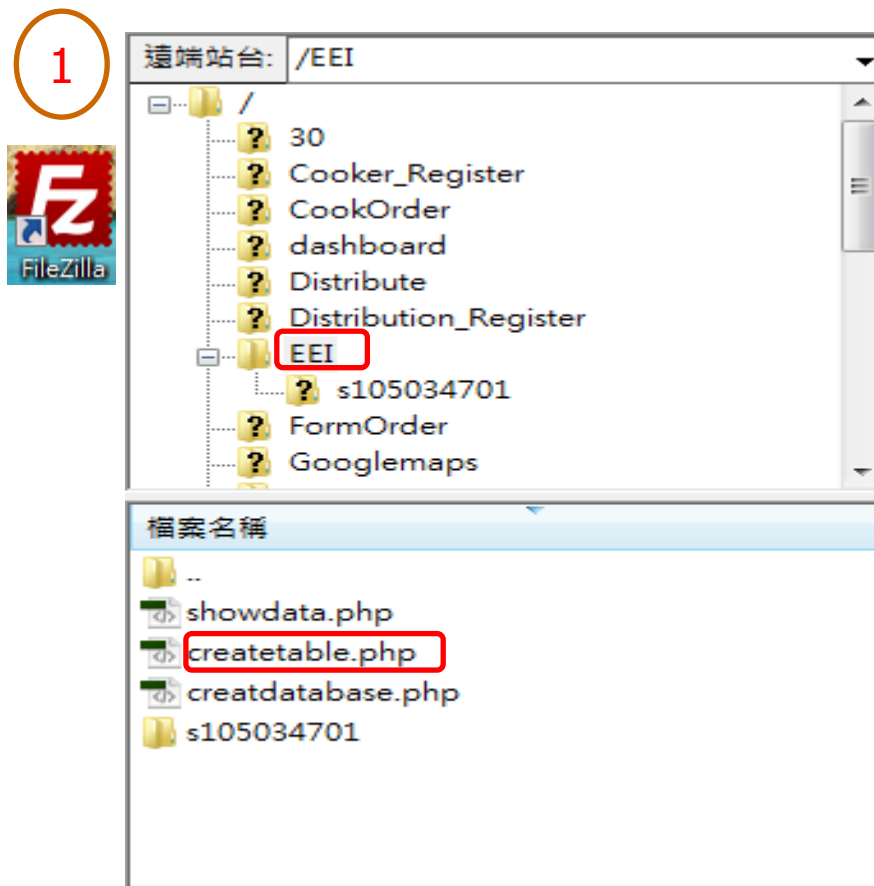
資料庫位置

資料表名稱

# Create table

```
<?php
define('HOST', 'localhost');
define('USER', '用戶名');
define('PASS', '密碼');
//輸入個人database位置
define('DB', 'EEI');
$con = mysqli_connect(HOST, USER, PASS, DB);
//連接資料庫所得到的資料庫皆由utf8 (unicode) 統一編碼
$con->query("SET NAMES 'utf8'");
//輸入需要的欄位項目以及table名稱
$sql = "CREATE TABLE s105034701 (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    uniid VARCHAR(30),
    name VARCHAR(30),
    product VARCHAR(50),
    kitchen VARCHAR(50),
    price VARCHAR(50),
    quantity VARCHAR(50),
    status VARCHAR(50),
    created_at datetime
)";
//成功、失敗訊息顯示
if ($con->query($sql) === TRUE) {
    echo "Success";
} else {
    echo "Error creating table:" . $con->error;
}
mysqli_close($con);
?>
```

# Create table





# 用PHP在資料表裡輸入值

- Inserting Values in tables

php: **mysqli\_connect** (主機名稱, 使用者, 密碼, 資料庫名稱)

sql: **INSERT INTO** 資料表名稱 (欄位) **VALUES** (資料)

```
<?php
$con=mysqli_connect("example.com","username","password","my db");
$sql="INSERT INTO table1 (FirstName, LastName, Age) VALUES ('admin', 'admin', 'administrator')";
if (mysqli_query($con,$sql)) { 資料表欄位          資料表欄位值
    echo "Values have been inserted successfully";
}
?>
```

# \$ \_GET指令與\$ \_POST指令

- GET and POST methods

```
<?php
$con=mysqli_connect("example.com","username","password","database name");

if (mysqli_connect_errno($con)) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$username = $_GET['username'];
$password = $_GET['password'];
$result = mysqli_query($con,"SELECT Role FROM table1 where Username='$username'
    and Password='$password'");
$row = mysqli_fetch_array($result);
$data = $row[0];

if($data){
    echo $data;
}
mysqli_close($con);
?>
```

# PHP & MySQL物件導向應用

- |           |        |
|-----------|--------|
| 1. 設定共用常數 | define |
| 2. 查詢     | SELECT |
| 3. 新增     | INSERT |
| 4. 更新     | UPDATE |
| 5. 刪除     | DELETE |

# PHP & MySQL物件導向應用

- 設定連結DB的共用常數 (db\_configs.php)

```
<?php  
define("host", "localhost", false); 主機位置  
define("username", "root", false); 使用者  
define("password", "mysql", false); 密碼  
define("dbname", "test", false); 資料庫名稱  
?>
```

# PHP & MySQL物件導向應用

- 查詢作業 (query\_ext.php)

```
<?php
// 設定文件utf-8編碼
header("Content-Type:text/html; charset=utf-8");
// 加入DB共用常變數
require_once '../db_configs.php';

$mysqli = new mysqli(host, username, password, dbname);

// 檢查連線總狀
if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli->connect_error;
}

// 設定MySQL為utf8編碼
$mysqli->set_charset("utf8");

// 查詢user_info資料表所有記錄
$sql = "SELECT * FROM user_info";
$result = $mysqli->query($sql);

echo "<h2>查詢範例(object)</h2>";
echo "=====";
echo "<table border='1'>
<tr>
<th>Username</th>
<th>Password</th>
</tr>";

while($row = $result->fetch_array()) {
    echo "<tr>";
    echo "<td>" . $row['username'] . "</td>";
    echo "<td>" . $row['password'] . "</td>";
    echo "</tr>";
}

echo "</table>";

// 關閉MySQL資料庫連線
$mysqli->close();
?>
```

# PHP & MySQL物件導向應用

- 新增作業 (add\_ext.php)

```
<?php
// 設定文件utf-8編碼
header("Content-Type:text/html; charset=utf-8");
// 加入DB共用常數
require_once '../db_configs.php';

$mysqli = new mysqli(host, username, password, dbname);

// 檢查連線狀態
if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli->connect_error;
}

// 設定MySQL為utf8編碼
$mysqli->set_charset("utf8");

// 新增一筆記錄到user_info資料表
$sql = "INSERT INTO user_info (username, password, birth_date, sex, age) VALUES ('Eric','eric12345','1995-02-24',1,19)";
$mysqli->query($sql);

echo "<h2>新增範例(object)</h2>";
echo "=====<br/>";
echo '新增一筆記錄成功!!';

// 關閉MySQL資料庫連線
$mysqli->close();
?>
```

# PHP & MySQL物件導向應用

- 更新作業 (update\_ext.php)

```
<?php
// 設定文件utf-8編碼
header("Content-Type:text/html; charset=utf-8");
// 加入DB共用常數
require_once '../db_configs.php';

$mysqli = new mysqli(host, username, password, dbname);

// 檢查連線總狀
if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli->connect_error;
}

// 設定MySQL為utf8編碼
$mysqli->set_charset("utf8");

// 更新某一筆記錄到user_info資料表
$sql = "UPDATE user_info SET password='pt54321' WHERE user_id=1";
$mysqli->query($sql);

echo "<h2>更新範例(object)</h2>";
echo "=====<br/>";
echo '更新user_id=1的記錄成功!!';

// 關閉MySQL資料庫連線
$mysqli->close();
?>
```

# PHP & MySQL物件導向應用

- 刪除作業 (delete\_ext.php)

```
<?php
// 設定文件utf-8編碼
header("Content-Type:text/html; charset=utf-8");
// 加入DB共用常數
require_once '../db_configs.php';

$mysqli = new mysqli(host, username, password, dbname);

// 檢查連線總狀
if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli->connect_error;
}

// 設定MySQL為utf8編碼
$mysqli->set_charset("utf8");

// 刪除某一筆記錄
$sql = "DELETE FROM user_info WHERE username='Eric'";
$mysqli->query($sql);

echo "<h2>刪除範例(object)</h2>";
echo "=====  
<br/>";
echo "刪除username='Eric'的記錄成功!!";

// 關閉MySQL資料庫連線
$mysqli->close();
?>
```



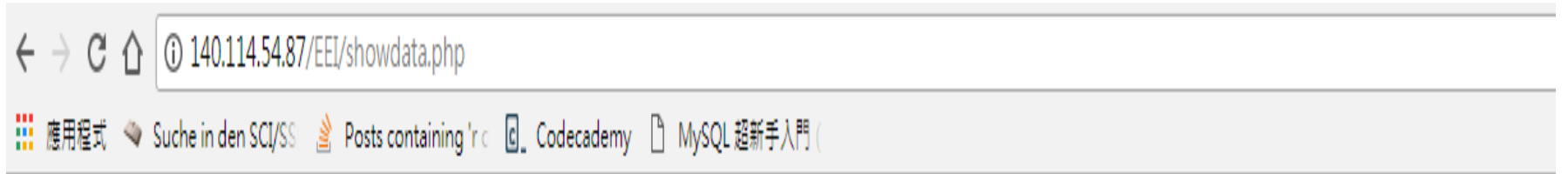
# Delete data

```
<?php
    $DB_USER='使用者名稱';
    $DB_PASS='密碼';
    $DB_HOST='localhost';
    //一樣為你要搜尋的資料庫
    $DB_NAME='ordermeal';
    $mysqli = new mysqli($DB_HOST, $DB_USER, $DB_PASS, $DB_NAME);
    /* check connection */
    if (mysqli_connect_errno()) {
        printf("Connect failed: %s\n", mysqli_connect_error());
        exit();
    }
    //搜尋時判斷所需要的回應，也就是她會根據接收到的$unique來搜尋資料庫
    $unique=$_REQUEST['unique'];
    $mysqli->query("SET NAMES 'utf8'");
    //sql刪除row語法
    $sql="DELETE FROM personorder WHERE uniid = '$unique'";
    //執行sql語法
    $result=$mysqli->query($sql);
    if($r=$result)
    {
        $flag['code']=1;
        echo "success";
    }
}
?>
```

# Show data

```
<?php
    define ('HOST', 'localhost');
    define ('USER', '使用者名稱');
    define ('PASS', '密碼');
    //database位置
    define ('DB', 'EEI');
    $mysqli = mysqli_connect (HOST, USER, PASS, DB);
    $mysqli->query ("SET NAMES 'utf8'");
    //選擇搜尋的table位置以及搜尋項目 (*為全部欄位皆搜尋)
    $sql="SELECT * FROM s105034701";
    $result=$mysqli->query ($sql);
    //把搜尋結果$e存入$output
    while ($e=mysqli_fetch_assoc ($result)) {
        $output []=$e;
    }
    //把$output轉換成json格式
    print (json_encode ($output));
    //關閉已開啟的資料庫
    $mysqli->close ();
?>
```

# Show data



```
[{"id": "1", "account": "", "pwd": "", "phone": "", "uniid": "dfaqr", "name": "qfqr", "product": "qfqrqew", "kitchen": "wegwrg", "price": "qegqeg", "quantity": "weg", "status": "te4trt", "created_at": "2017-09-21 00:00:00"}, {"id": "2", "account": "gwrq", "pwd": "wrgwq", "phone": "wggwe", "uniid": null, "name": null, "product": null, "kitchen": null, "price": null, "quantity": null, "status": null, "created_at": null}]
```

phpMyAdmin

	id	account	pwd	phone	uniid	name	product	kitchen	price	quantity	status	created_at
<input type="checkbox"/> 编辑 复制 删除	1				dfaqr	qfqr	qfqrqew	wegwrg	qegqeg	weg	te4trt	2017-09-21 00:00:00
<input type="checkbox"/> 编辑 复制 删除	2	gwrq	wrgwq	wggwe	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

# phpMyadmin

- <http://140.114.54.94/phpmyadmin/>



The image shows the phpMyAdmin login page. At the top, there is a logo for phpMyAdmin featuring a sailboat. Below the logo, the text "歡迎使用 phpMyAdmin" (Welcome to use phpMyAdmin) is displayed. The page is divided into two main sections. The first section, titled "語言 - Language", contains a dropdown menu with "中文 - Chinese traditional" selected. The second section, titled "登入" (Login), contains two input fields: "使用者名稱:" (Username) with the value "2020iie" and "密碼:" (Password) which is empty. At the bottom right of the login section, there is a button labeled "執行" (Execute).

帳號：2023iie  
密碼：2023

# phpMyadmin



The screenshot displays the phpMyAdmin web interface. At the top, the server address is 127.0.0.1. The navigation bar includes options for 資料庫 (Databases), SQL, 狀態 (Status), 使用者帳號 (User accounts), 匯出 (Export), and 匯入 (Import). The left sidebar shows a tree view of databases, including a newly added '2020iie' database with sub-databases '2020iie\_group1' through '2020iie\_group11', and 'information\_schema' and 'test'. The main content area is divided into two sections: '一般設定' (General) and '外觀設定' (Appearance). Under '一般設定', there is an option to '更改密碼' (Change password) and a dropdown for '伺服器連線編碼與排序' (Server connection encoding and collation) set to 'utf8mb4\_unicode\_ci'. Under '外觀設定', there is a dropdown for '語言 - Language' (Language) set to '中文 - Chinese traditional', a dropdown for '佈景主題' (Theme) set to 'pmahomme', and a dropdown for '字體大小' (Font size) set to '82%'. A '更多設定' (More settings) link is also visible.

# 建立資料庫



輸入資料庫名稱

選utf8\_unicode\_ci

# 建立資料表

The screenshot shows a database management tool interface. At the top, the address bar indicates the server is at 127.0.0.1 and the database is named 'database1'. Below the address bar is a menu bar with various options: 結構, SQL, 搜索, 查詢, 導出, 導入, 操作, 權限, 程序, 事件, 觸發器, 追踪, 設計器, and 文本框列. A message box below the menu bar states '數據庫中沒有表。' (No tables in the database). Below this, there is a '新建數據表' (New Table) button. Underneath, there is a form with two input fields: '名字:' (Name) containing 'table1' and '字段數:' (Number of fields) containing '4'. The '名字:' field is highlighted with a red rectangle. At the bottom right of the form, there is an '執行' (Execute) button.

數據庫中沒有表。

新建數據表

名字: table1 字段數: 4

輸入資料表名稱

執行

# 建立資料表

服務器: 127.0.0.1 » 數據庫: database1 » 表: table1

瀏覽 結構 SQL 搜索 插入 導出 導入 權限 操作 追踪

數據表名: table1 添加 1 column(s) 執行

欄位的名稱	指定長度	類型	默認	排序規則
名字	長度/值			
留言編號 Pick from Central Columns		INT	無	
姓名 Pick from Central Columns	20	VARCHAR	無	
留言 Pick from Central Columns	256	VARCHAR	無	
日期時間 Pick from Central Columns		DATETIME	無	

表註釋: 選擇欄位的資料型別

Collation:

儲存引擎: InnoDB

儲存引擎在此採用預設值即可



# 主鍵設定



服務器: 127.0.0.1 \* 數據庫: database1 \* 表: table1

瀏覽 結構 SQL 搜索 插入 導出 導入 權限 操作 追踪

表結構 關聯視圖

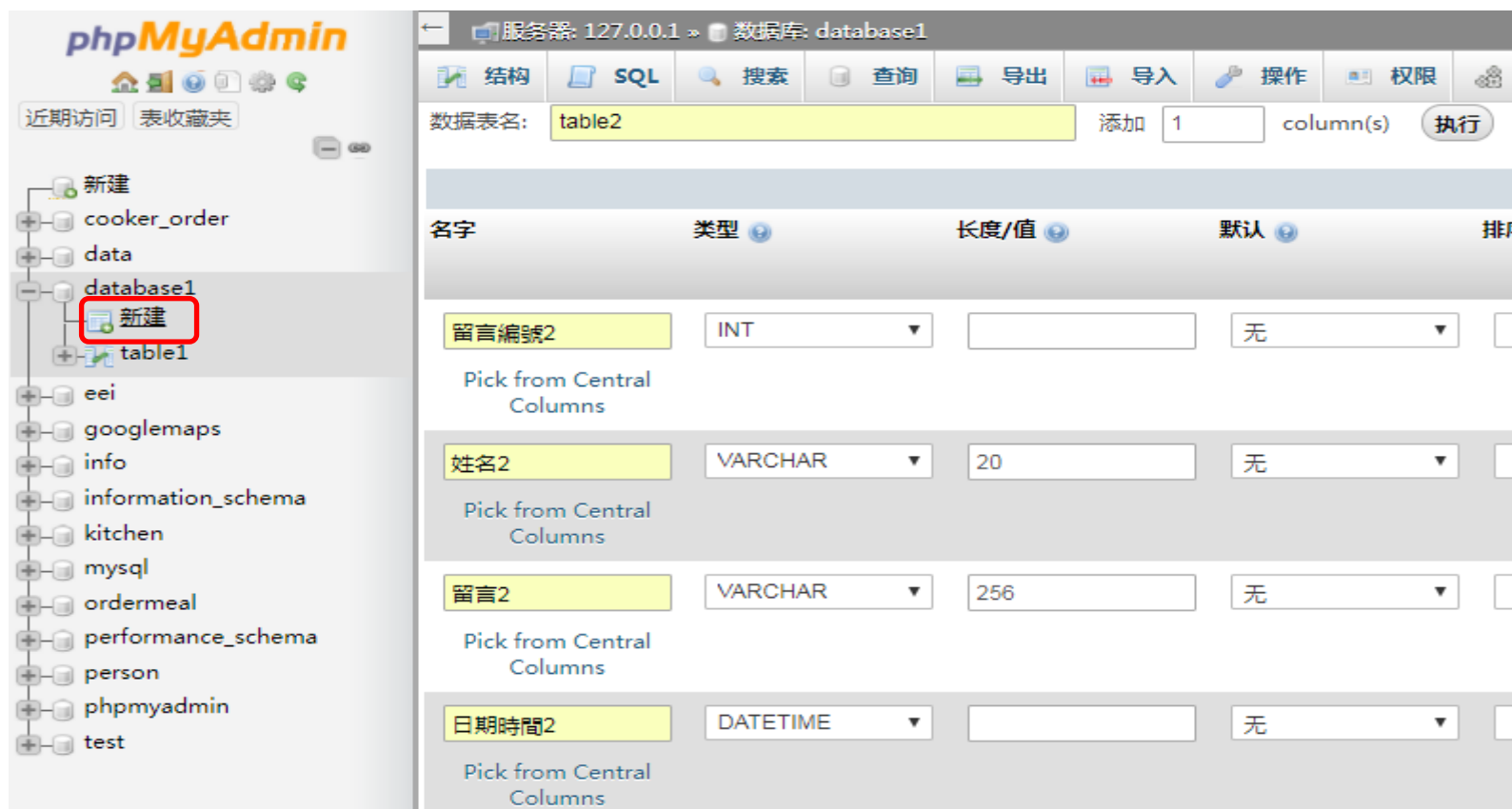
#	名字	類型	排序規則	屬性	空	默認	註釋	額外	操作
<input type="checkbox"/>	1 留言編號	int(11)			否	無			修改 刪除 <b>主鍵</b> 唯一 索引
<input type="checkbox"/>	2 姓名	varchar(20)	latin1_swedish_ci		否	無			修改 刪除 主鍵 唯一 索引
<input type="checkbox"/>	3 留言	varchar(256)	latin1_swedish_ci		否	無			修改 刪除 主鍵 唯一 索引
<input type="checkbox"/>	4 日期時間	datetime			否	無			修改 刪除 主鍵 唯一 索引

全選 選中項: 瀏覽 修改 刪除 主鍵 唯一 索引 Add to central columns

打印 規劃表結構 追踪表 移動字段 Improve table structure

添加 1 個字段 於日期時間之後 執行

# 新增資料表



The screenshot shows the phpMyAdmin interface. On the left sidebar, the 'database1' folder is expanded, and a '新建' (New) button is highlighted with a red box. The main panel shows the 'Structure' view for 'table2' in 'database1'. The 'Add' button is set to '1' column(s). Below the table structure, four columns are being added:

名字	类型	长度/值	默认	排序
留言編號2	INT		无	
姓名2	VARCHAR	20	无	
留言2	VARCHAR	256	无	
日期時間2	DATETIME		无	

# SQL 語法

服務器: 127.0.0.1

數據庫 SQL 狀態 帳戶 導出 導入 設置 複製 變量

在服務器"127.0.0.1" 運行SQL 查詢:

可利用SQL語法完成建立資料庫和資料表的動作

```
CREATE DATABASE `mytes` DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;

CREATE TABLE `guestbook` (
  `留言編號` INT UNSIGNED NOT NULL,
  `姓名` VARCHAR( 20 ) NOT NULL DEFAULT '無名士',
  `留言` VARCHAR( 256 ) NULL,
  `日期時間` DATETIME NOT NULL,
  PRIMARY KEY ( `留言編號` ) ENGINE = memory;
```

資料庫名稱

資料表名稱

清除 格式 Get auto-saved query

Bind parameters

Bookmark this SQL query:

[ 語句定界符 ; ]  在此再次顯示此查詢  保留查詢框  Rollback when finished  啟用外鍵約束

# 將數據插入資料表

← 服务器: 127.0.0.1 » 数据库: database1 » 表: table1

浏览 结构 SQL 搜索 插入 导出 导入 权限 操作

字段	类型	函数	空值
留言編號	int(11)	<input type="text"/>	<input type="text"/>
姓名	varchar(20)	<input type="text"/>	張豪展
留言	varchar(256)	<input type="text"/>	今天天氣真好
日期時間	datetime	<input type="text"/>	2017-09-21 10:28:47

執行

# SQL insert 語法



服务器: 127.0.0.1 » 数据库: database1 » 表: table1

浏览 结构 SQL 搜索 插入 导出 导入 权限 操作 追踪 触发器

插入了 1 行。  
Warning: #1366 Incorrect integer value: "" for column '留言編號' at row 1

```
INSERT INTO `table1` ('留言編號', '姓名', '留言', '日期時間') VALUES ('', '張豪展', '今天天氣真好', '2017-09-21 10:28:47');
```

[ 編輯內嵌 ] [ 編輯 ] [ 創建 PHP 代碼 ]

Run SQL query/queries on table database1.table1:

```
1 INSERT INTO `table1` ('留言編號', '姓名', '留言', '日期時間') VALUES ('', '張豪展', '今天天氣真好', '2017-09-21 10:28:47');
```

SELECT\* SELECT INSERT UPDATE DELETE 清除 格式 Get auto-saved query

Bind parameters

Bookmark this SQL query:

[ 语句定界符 ; ]  在此再次显示此查询  保留查询框  Rollback when finished  启用外键约束

執行

字段  
留言編號  
姓名  
留言  
日期時間

# 刪除資料表

The screenshot shows a database management tool interface for a server at 127.0.0.1 and a database named 'database1'. The main table list shows two tables: 'table1' and 'table2'. The 'DELETE' icon for 'table2' is highlighted with a red box. A confirmation dialog box is open, asking '您將要完全刪除一個表！您真的要執行 DROP TABLE table2 嗎？' (Are you sure you want to completely delete a table! Do you really want to execute DROP TABLE table2?). The 'DROP TABLE table2' text in the dialog is also highlighted with a red box. The dialog includes a checked checkbox for '啟用外鍵約束' (Enable foreign key constraints) and buttons for '確定' (OK) and '取消' (Cancel).

表	操作	行數	類型	排序規則	大小	多餘
table1	瀏覽 結構 搜索 插入 清空 刪除	2	InnoDB	utf8_unicode_ci	16 KB	-
table2	瀏覽 結構 搜索 插入 清空 刪除	0	InnoDB	utf8_unicode_ci	16 KB	-
2 張表 總計		2	InnoDB	utf8_unicode_ci	32 KB	0 字節

# 資料庫備份

- 分為兩種層級：
  1. 資料庫備份
  2. 資料表備份



# 資料庫備份



伺服器: 127.0.0.1 » 資料庫: database1

結構 SQL 搜索 查詢 導出 導入 操作 權限 程序 事件 觸發器 追踪 設計器 文本框列

## 正在導出數據庫 "database1" 中的數據表

導出模板:

新模板: 現有模板:

模板名称   模板:

導出方式:

- 快速- 顯示最少的選項
- 自定義- 顯示所有可用的選項

格式:

SQL



# 資料庫備份

```
--  
-- Database: `database1`  
--  
-----  
  
--  
-- 表的结构 `table1`  
--  
--  
CREATE TABLE `table1` (  
  `留言編號` int(11) UNSIGNED NOT NULL,  
  `姓名` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `留言` varchar(256) COLLATE utf8_unicode_ci NOT NULL,  
  `日期時間` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;  
  
--  
-- 转存表中的数据 `table1`  
--  
--  
INSERT INTO `table1` (`留言編號`, `姓名`, `留言`, `日期時間`) VALUES  
(0, '張豪展', '今天天氣真好', '2017-09-21 10:28:47'),  
(1, '張豪展', '中午吃什麼', '2017-09-21 12:30:23');  
  
-----  
  
--  
-- 表的结构 `table2`  
--  
--  
CREATE TABLE `table2` (  
  `留言編號2` int(10) UNSIGNED NOT NULL,  
  `姓名2` varchar(20) COLLATE utf8_unicode_ci NOT NULL,  
  `留言2` varchar(265) COLLATE utf8_unicode_ci NOT NULL,  
  `日期時間2` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

# 資料庫備份

```
-- Database: `database1`
-----
-- 表的结构 `table1`
CREATE TABLE `table1` (
  `留言編號` int(11) UNSIGNED NOT NULL,
  `姓名` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `留言` varchar(256) COLLATE utf8_unicode_ci NOT NULL,
  `日期時間` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- 转存表中的数据 `table1`
INSERT INTO `table1` (`留言編號`, `姓名`, `留言`, `日期時間`) VALUES
(0, '張豪展', '今天天氣真好', '2017-09-21 10:28:47'),
(1, '張豪展', '中午吃什麼', '2017-09-21 12:30:23');
-----
-- 表的结构 `table2`
CREATE TABLE `table2` (
  `留言編號2` int(10) UNSIGNED NOT NULL,
  `姓名2` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `留言2` varchar(265) COLLATE utf8_unicode_ci NOT NULL,
  `日期時間2` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

資料庫名稱

資料表名稱與資料表結構

資料表裡的數據和內容

# 導入資料表



← 服務器: 127.0.0.1 » 數據庫: database1

結構 SQL 搜索 查詢 導出 導入 操作 權限 程序 事件 觸發器

## 導入到數據庫 "database1"

要導入的文件：

文件可能已壓縮(gzip, bzip2, zip)或未壓縮。  
壓縮文件名必須以.[格式].[壓縮方式]結尾。如：.sql.zip

從計算機中上傳： **選擇檔案** database1 (2).sql (最大限制：2,048 KB) 您可在各頁面間拖放文件。

文件的字符集： utf-8 database1 (2).sql

部分導入：

在導入時腳本若檢測到可能需要花費很長時間（接近PHP超時的限定）則允許中斷。（儘管這會中斷事務，但在導入大文件時是個很好的方法。）

Skip this number of queries (for SQL) starting from the first one:

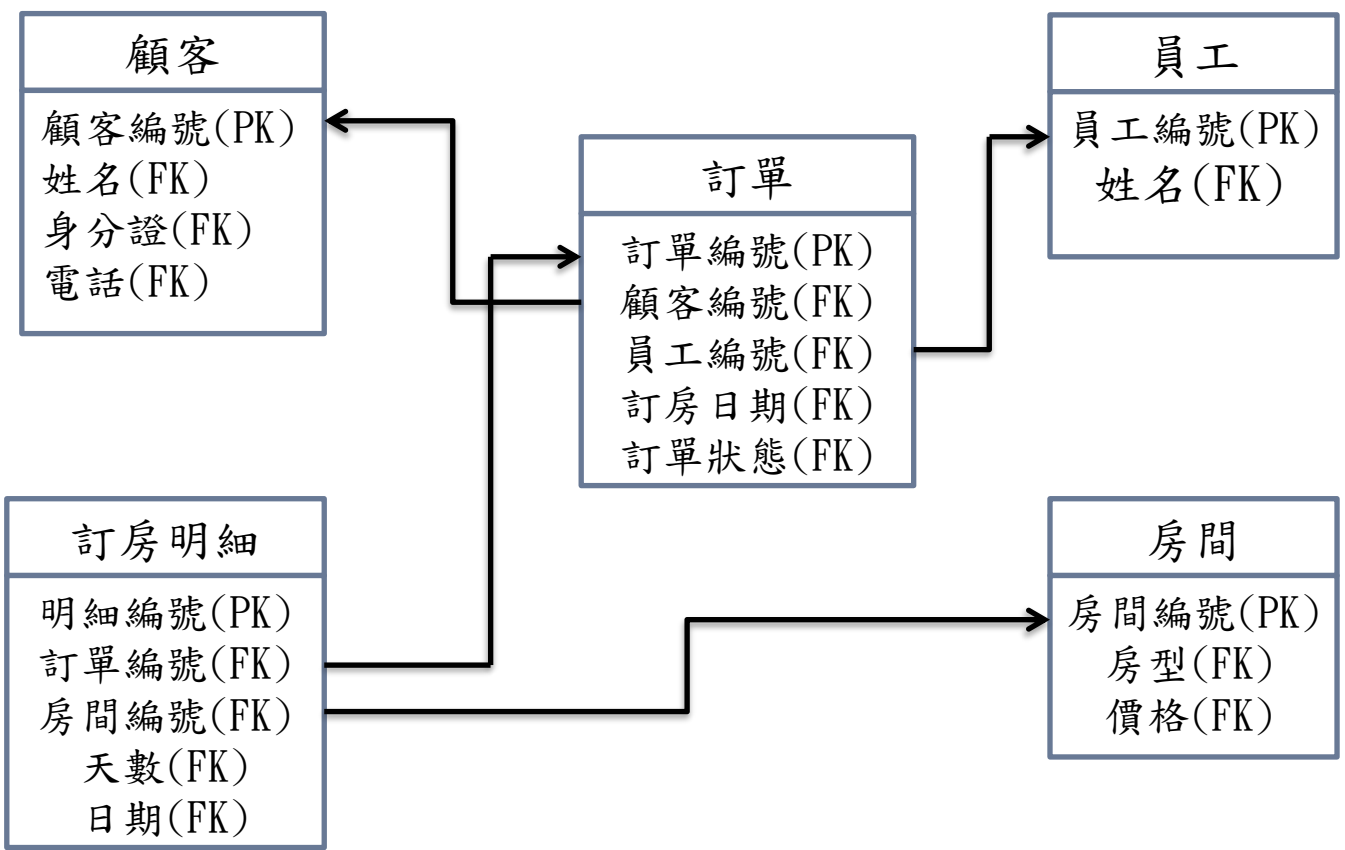
# 導入資料表

The screenshot shows the phpMyAdmin interface for a database named 'database1'. The left sidebar shows a tree view of the database structure, with 'table2' highlighted in a red box. The main area displays a table list with columns: 表 (Table), 操作 (Operations), 行数 (Rows), 类型 (Type), 排序规则 (Collation), 大小 (Size), and 多余 (Extra). The table list contains two tables: 'table1' and 'table2'. 'table2' is highlighted in a red box. Below the table list, there is a '全选' (Select all) button and a '选中项' (Selected items) dropdown menu. At the bottom, there are '打印' (Print) and 'Data dictionary' buttons.

表	操作	行数	类型	排序规则	大小	多余
table1	浏览 结构 搜索 插入 清空 删除	2	InnoDB	utf8_unicode_ci	16 KB	-
table2	浏览 结构 搜索 插入 清空 删除	0	InnoDB	utf8_unicode_ci	16 KB	-
2 张表 总计		2	InnoDB	utf8_unicode_ci	32 KB	0 字节

# 隨堂作業

將以下ERD轉成phpmyadmin的資料表



## 回家作業

1. 於自己組別的资料庫中，新增一資料表  
名稱為 **s+學號** 例如 s108034539  
(資料表的欄位內容、資料型態規定在下一頁ppt)
2. 在資料表中，插入一筆完整的資料(自行假設)

使用phpmyadmin

主機：<http://140.114.54.94/phpMyAdmin/>

帳號：2023iie

密碼：2023

# 回家作業

欄位名稱	資料型態	長度	主索引鍵	說明
id	INT	-	✓	自動編號(AI)(auto_increment)
account	VARCHAR	10		帳號
password	VARCHAR	10		密碼
name	VARCHAR	10		姓名
sex	CHAR	2		性別
year	TINYINT	-		出生年
month	TINYINT	-		出生月
day	TINYINT	-		出生日
telephone	VARCHAR	20		電話
cellphone	VARCHAR	20		行動電話
address	VARCHAR	50		地址
email	VARCHAR	50		電子郵件
url	VARCHAR	50		個人網址
comment	TEXT	-		備註欄位

## 參考網站

- CodeData

<http://www.codedata.com.tw/database/mysql-tutorial-getting-started>

- 傑瑞學習分享

<http://jerry17768java.blogspot.tw/2014/08/php.html>

- Tutorialspoint

[https://www.tutorialspoint.com/android/android\\_php\\_mysql.htm](https://www.tutorialspoint.com/android/android_php_mysql.htm)

- PHP: MySQL-Manual

<http://php.net/manual/en/book.mysql.php>



**Thank you for listening!!**