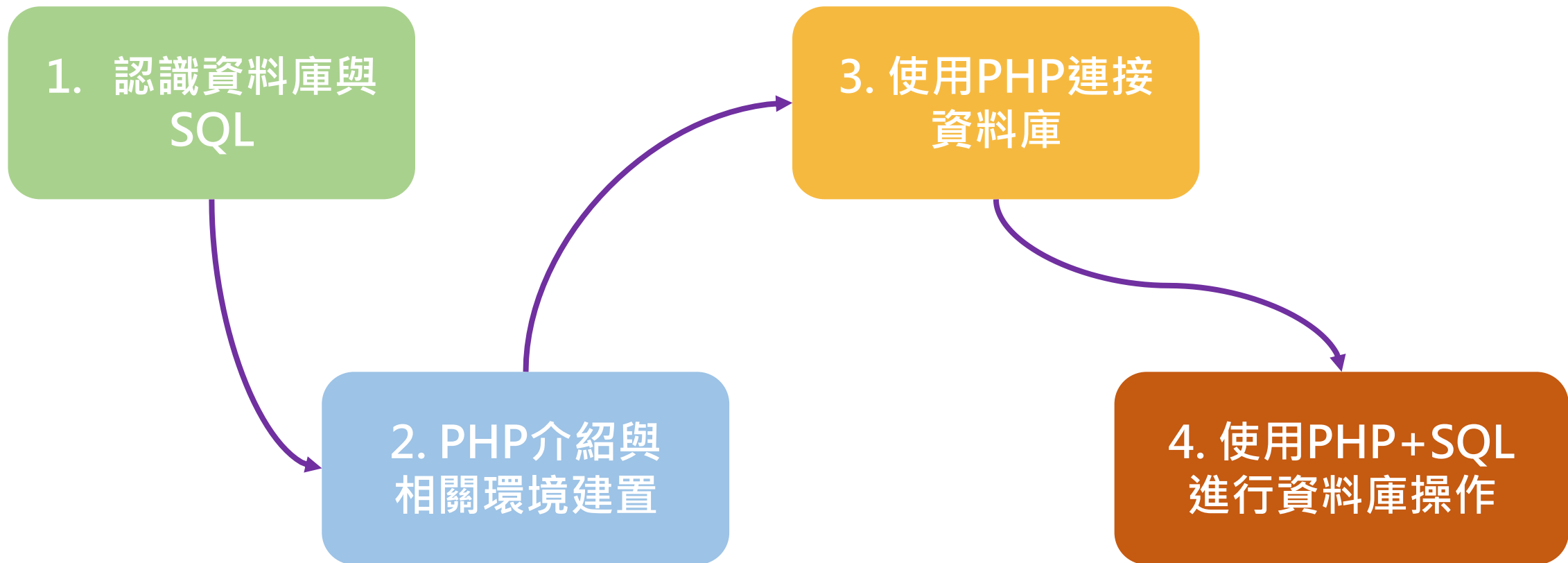


教育數據分析

SDPMLab, NKTU
軟體開發與流程管理實驗室
國立高雄師範大學

課程大綱



課程檔案下載

<https://shorturl.at/gluzC>

本章節簡報於lesson 1 資料夾中。

認識資料庫與SQL

SDPMLab, NKTU
軟體開發與流程管理實驗室
國立高雄師範大學

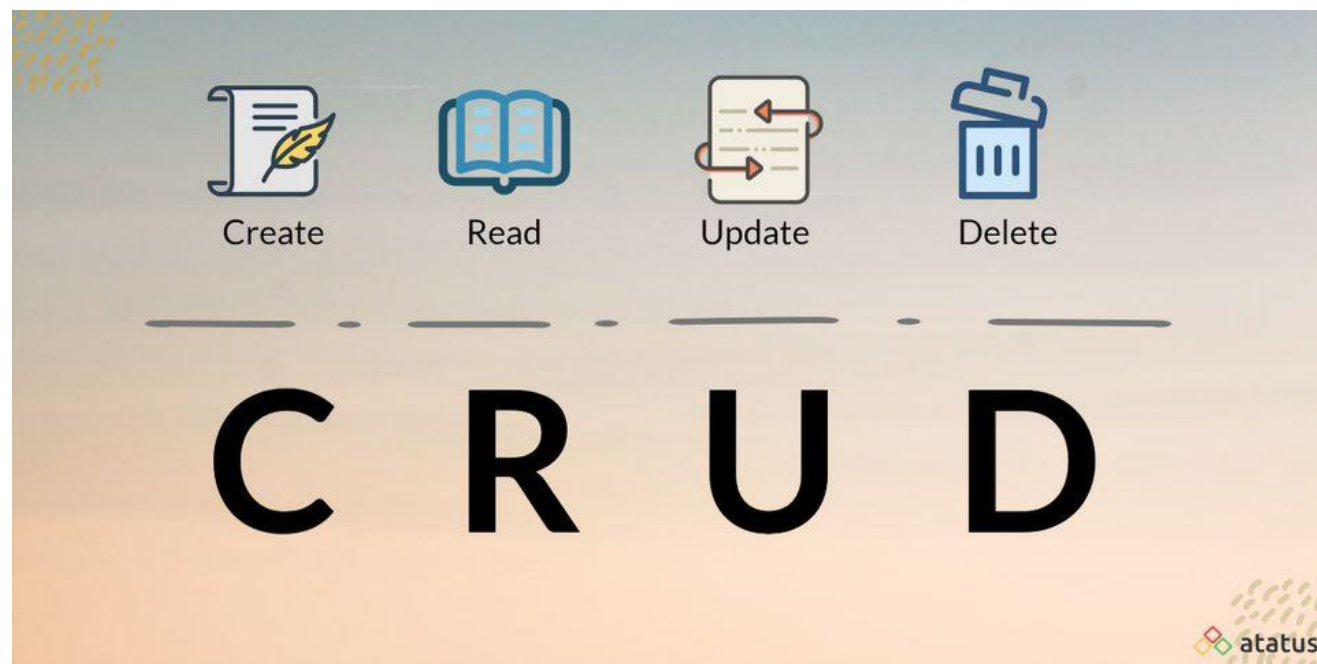
認識SQL與資料庫 – 什麼是資料庫？

1. 資料庫 (**Database**) 是一個組織化的數據集合，可**用於存儲、檢索和管理數據**。
2. 常見應用：網站、電商平台、社交平台、金融服務。
3. 資料庫類型：
 - 關聯型資料庫 (**RDBMS**)：基於關聯模型，數據以表格的形式存儲。例如：MySQL，Oracle，SQLServer。
 - 非關聯型資料庫 (**NoSQL**)：不僅限於關聯模型，適用於大數據和實時應用程序。例如：MongoDB，Redis。
4. 組成元件：
 - 表 (**Tables**)：存儲數據的地方，由行和列組成。
 - 字段 (**Fields**) / 列 (**Columns**)：表中的一列，存儲特定類型的數據。
 - 記錄 (**Records**) / 行 (**Rows**)：表中的一行，包含一條特定的信息。

認識SQL與資料庫 - 資料庫操作

常見的資料庫操作：**CRUD**

- **C** (Create): 創建新數據。
- **R** (Read): 讀取/檢索數據。
- **U** (Update): 更新數據。
- **D** (Delete): 刪除數據。



圖片來源：<https://www.atatus.com/glossary/content/images/size/w960/2021/07/CRUD.jpeg>

認識SQL與資料庫 - 什麼是SQL？

1. 全名：Structured Query Language，簡稱**SQL**，為一種資料庫查詢和程式設計語言。
2. 用途：用於存取資料以及查詢、更新和管理關聯式資料庫系統。
3. 使用場景：於系統中使用到的CRUD，皆會使用到SQL。



圖片來源：<https://usa.bootcampcdn.com/wp-content/uploads/sites/106/2020/03/SQL-Coding-Class-San-Francisco-1.jpeg>

認識SQL與資料庫 - 常用的SQL關鍵字

資料庫(Database)的創建和刪除

- 創建資料庫

```
sql Copy code  
CREATE DATABASE database_name;
```

- 刪除資料庫

```
sql Copy code  
DROP DATABASE database_name;
```

以**創建資料庫為例**，其中的**CREATE**與**DROP**關鍵字為針對資料庫的操作，**CREATE DATABASE**為一個組合語法，接續的database_name參數則為欲建立的資料庫名稱。

認識SQL與資料庫 - 常用的SQL關鍵字

表(Tables)的創建、修改和刪除

- 創建資料表

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ...  
);
```

- 刪除資料表

```
DROP TABLE table_name;
```

- 修改資料表(以新增欄位為例)

```
ALTER TABLE table_name  
ADD column_name datatype;
```

以創建資料表為例，對於**資料表的操作**，則是將**CREATE加上TABLE**，形成一個組合語法，再加入欲新增的資料表名稱**table_name**，並以括號“ () ”包裹欲新增的資料欄位並賦予欄位型別，如文字(TEXT)、日期 (DATE) 等...。

認識SQL與資料庫 - 常用的SQL關鍵字

對資料的操作(新增、更新和刪除)

- 新增資料

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

- 更新資料

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

- 刪除資料

```
DELETE FROM table_name WHERE condition;
```

- 新增資料，以 **INSERT INTO** 作為關鍵字加上要操作的 **資料表名稱**，並指定 **欲新增資料的欄位**，使用 **VALUES** 關鍵字進行區隔再輸入 **欲新增的真實資料**。
- 更新資料，以 **UPDATE** 作為關鍵字加上要操作的 **資料表名稱**，並使用 **SET**，將 **資料欄位與欲更改的資料進行賦值**，當然可以使用 **WHERE** 進行條件限制。
- 刪除資料，以 **DELETE FROM** 作為關鍵字，加上要操作的 **資料表名稱**，並使用 **WHERE** 宣告需要 **被刪除之資料的條件限制**。

認識SQL與資料庫 - 常用的SQL關鍵字

資料的查詢

- 查詢所有欄位

```
SELECT * FROM table_name;
```

- 查詢特定欄位

```
SELECT column1, column2, ...  
FROM table_name;
```

- 加入條件的查詢

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

資料搜尋，主要使用**SELECT**關鍵字，
SELECT後接續的參數可以是 ***** (意指全部欄位) 或 **欄位名稱**，透過**FROM**進行資料表的指定，如搜尋是具有條件性的搜尋，則可**加入WHERE進行條件限制**。

認識SQL與資料庫 - 常用的SQL關鍵字

排序所查詢的資料

- 排序

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

前半段語法與一般搜尋無異，排序行為須用到 **ORDER BY** 關鍵字，透過 **ORDER BY** 進行欄位指定，意指以哪個欄位進行資料排序，排序分別可以

- 由小至大 (**Ascending**; 預設)
- 由大至小 (**Descending**)

認識SQL與資料庫 - 常用的SQL關鍵字

聚合資料查詢

- GROUP BY

```
SELECT column_name(s), aggregate_function(column_name)
FROM table_name
WHERE condition
GROUP BY column_name(s);
```

- EXAMPLE

資料

O_Id	Price	Customer
1	1000	張一
2	2000	王二
3	500	李三
4	1300	張一
5	1800	王二

前半段語法與一般搜尋無太大差異，主要會搭配**聚合函數**進行搜尋，排序行為須用到 **GROUP BY** 關鍵字，透過 GROUP BY 進行**欄位指定**，意指使用哪個欄位來進行分組，而每個群組都會回傳一個資料列，因為在一般搜尋中欄位中的資料可能存在重複值，以該值進行群組集結。

SQL

```
SELECT Customer, SUM(Price) FROM orders
GROUP BY Customer;
```

結果

Customer	SUM(Price)
張一	2300
王二	3800
李三	500

認識SQL與資料庫 – 常用的SQL關鍵字

常見聚合函數 (aggregate_function)

- COUNT() 主要用於計算**資料筆數**。
- SUM() 主要用於計算**資料總和**。
- MAX() 主要用於計算**資料最大值**。
- MIN() 主要用於計算**資料最小值**。
- AVG() 主要用於計算**資料平均值**。
- DISTINCT 主要用於取得不重複資料。

EXAMPLE

```
SELECT Customer, SUM(Price) FROM orders  
GROUP BY Customer;
```

PHP介紹與 相關環境建置

SDPMLab, NKTU
軟體開發與流程管理實驗室
國立高雄師範大學

PHP介紹與相關環境建置 – 什麼是PHP?

1. PHP (Hypertext Preprocessor)，是一種開源的伺服器端腳本語言。
2. 用途：主要用於網站或伺服器端開發，可嵌入HTML代碼中執行。
3. 特點：
 - 可跨平台運行。
 - 支持多種資料庫，如MySQL、PostgreSQL等。
 - 與HTML緊密結合，方便網頁創建和維護。
4. 本次教學主要使用到**PHP進行資料庫連接**，並**使用SQL**對資料庫**進行特定資料搜尋**。
5. PHP為單執行緒語言，撰寫方式與C語言和Perl類似，需用**<?php ?>**將程式碼包裹。

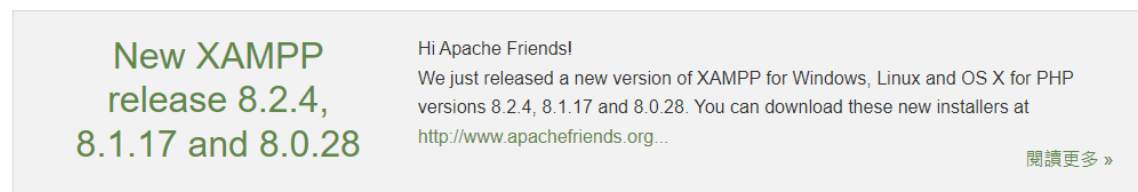
PHP介紹與相關環境建置 – 建立環境

1. 接下來的教學將使用 **XAMPP** 軟體，該軟體可使我們快速建置PHP網頁伺服器與資料庫環境。
2. XAMPP是一個將 Apache 網頁伺服器與 PHP、Perl 及 MariaDB 集合在一起的安裝包，允許使用者可以在自己的電腦上輕易的建立**網頁伺服器**。
3. XAMPP 官方網站與下載點：https://www.apachefriends.org/zh_tw/index.html

PHP介紹與相關環境建置 - 建立環境

STEP 1 .

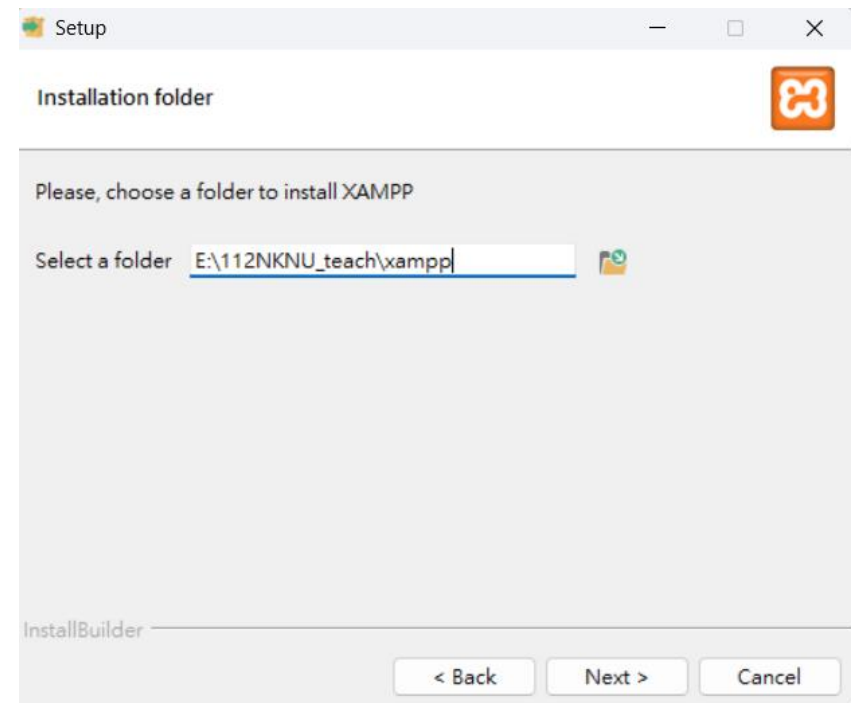
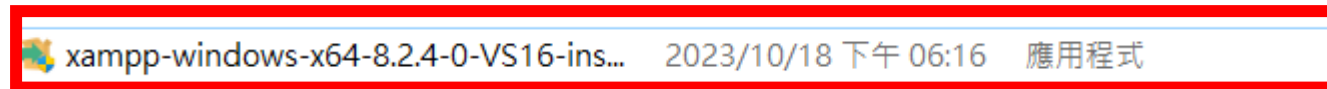
- 點擊上頁簡報中的XAMPP連結後，你將進入右方頁面，此時，請選擇與你電腦系統匹配的載點。
- 點擊後，稍待片刻XAMPP下載執行檔將下載至電腦中。



PHP介紹與相關環境建置 - 建立環境

STEP 2 .

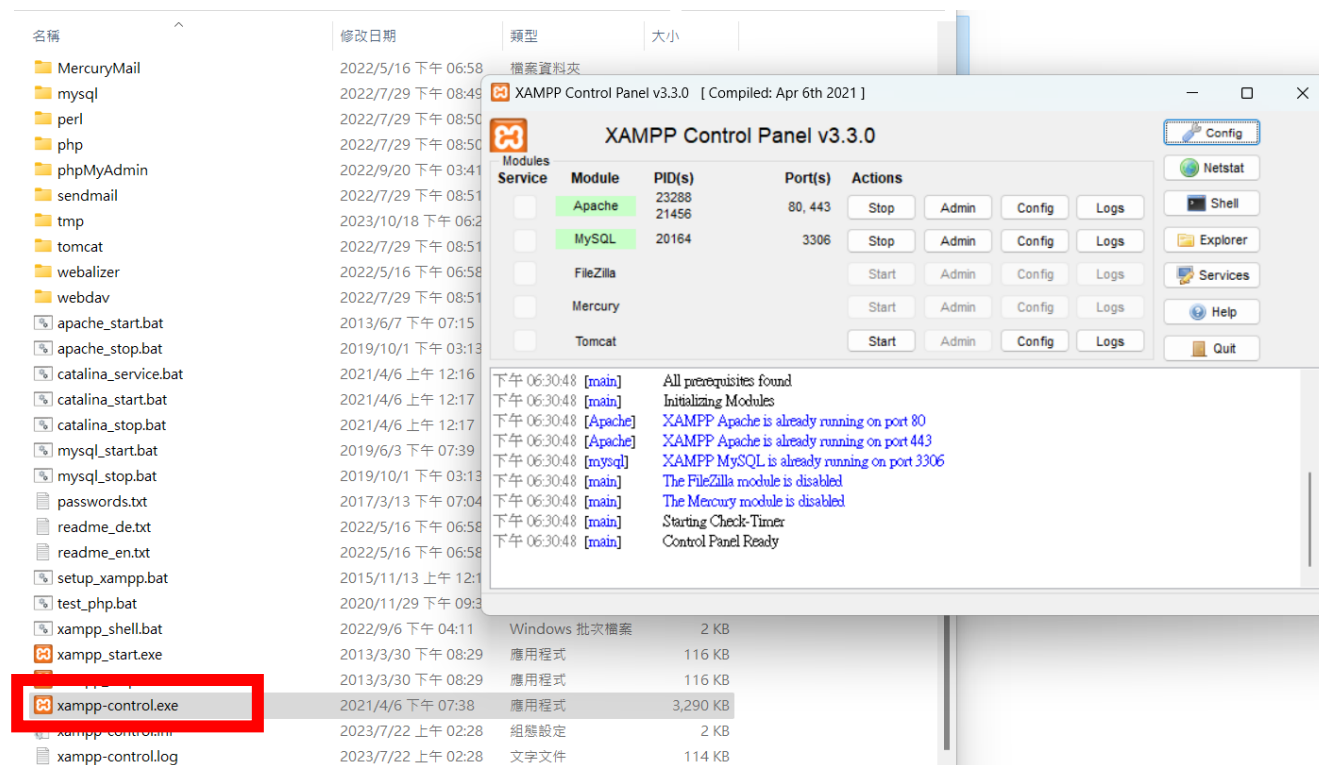
- 下載完畢後，你將得到右上圖中的執行檔，此時請執行該檔案。
- 執行後，將會開啟下載視窗，此時連續按下一步即可。
- 如遭遇 “The selected folder is not empty. Please select a different folder” 時，則於任一資料夾中建立名為xampp之資料夾，並將其指定為xampp安裝路徑，接下來繼續即可。



PHP介紹與相關環境建置 - 建立環境

STEP 3 .

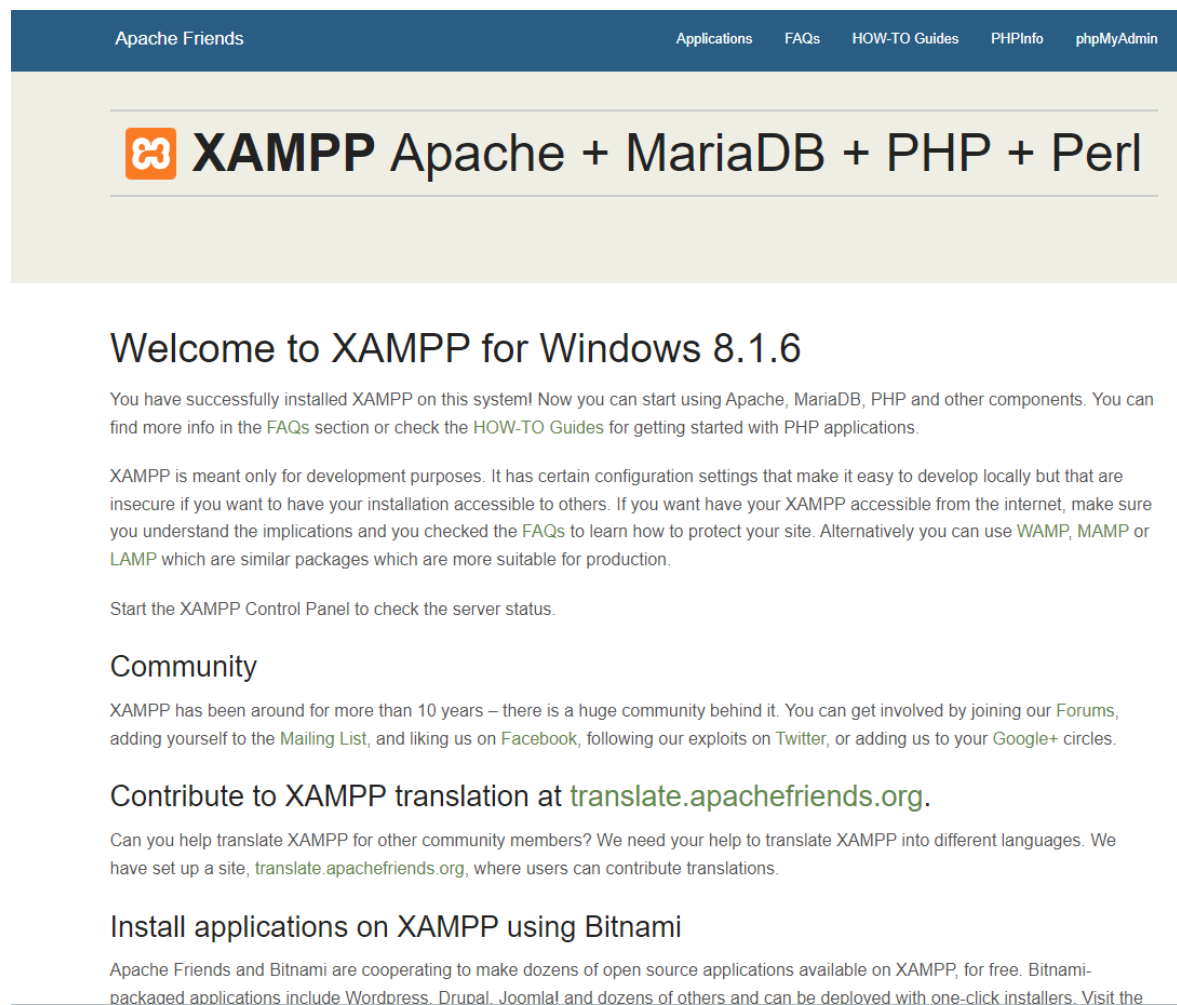
- 下載完畢後，請開啟安裝XAMPP的資料夾，進入目錄後找尋檔案名為 “**xampp-control.exe**” 的執行檔並執行。
- 執行後，將會出現如右圖之視窗。
- 將Apache 與 MySQL開啟。



PHP介紹與相關環境建置 - 建立環境

STEP 4 .

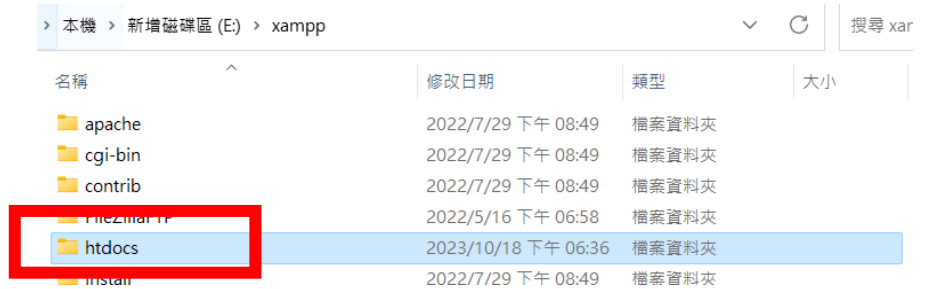
- 開啟瀏覽器並於搜尋欄鍵入 **localhost** 或 **127.0.0.1**，查看是否可看到與右圖相符的畫面。



PHP介紹與相關環境建置 – 建立環境

STEP 5 .

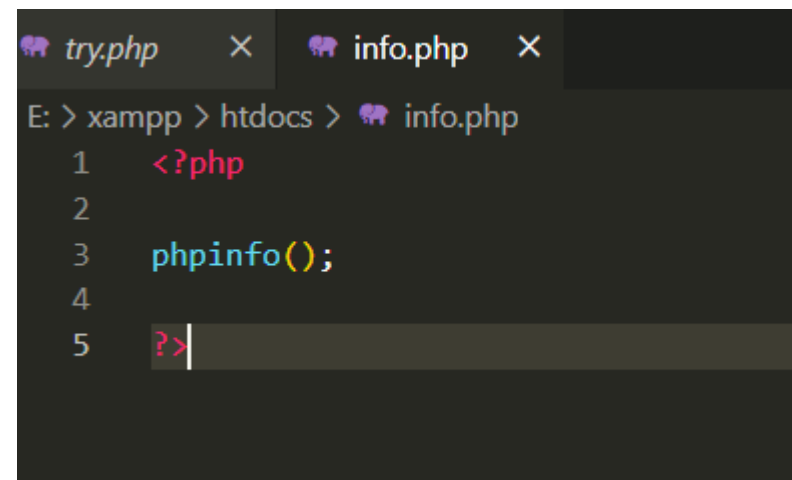
- 於XAMPP安裝目錄找尋 “**htdocs**” 目錄並進入，該目錄主要讓我們放置欲直行的php檔案。
- 建立一支PHP檔案，名為 “**info.php**”
- 使用任何習慣的編輯器，開啟該PHP檔案。



PHP介紹與相關環境建置 - 建立環境

STEP 6 .

- 於info.php中開始撰寫程式。
- 遵循PHP規則，使用`<?php ?>` 將程式碼包裹。
- 請遵循右圖於程式碼內新增一行“ `phpinfo();`” 程式碼，這將能讓你待會更好的了解已安裝的PHP擴展與檔案路徑關係。



```
try.php x info.php x
E: > xampp > htdocs > info.php
1  <?php
2
3  phpinfo();
4
5  ?>
```

PHP介紹與相關環境建置 - 建立環境

STEP 7 .

- 於瀏覽器搜尋框中鍵入 **127.0.0.1/info.php** 或 **localhost/info.php**，此時你將可看到與右圖相同的頁面。
- 此時網頁伺服器已建置完成。

127.0.0.1/info.php

PHP Version 8.1.6



Configure Command	cscrip /nologo /e:jscrip configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=.\..\..\instantclient\shared" "--with-oci8-19=.\..\..\instantclient\shared" "--enable-object-out-dir=.obj" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	E:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.TS.VS16
PHP Extension Build	API20210902.TS.VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.1.6, Copyright (c) Zend Technologies

zend engine

PHP介紹與相關環境建置 – 建立環境

STEP 8 .

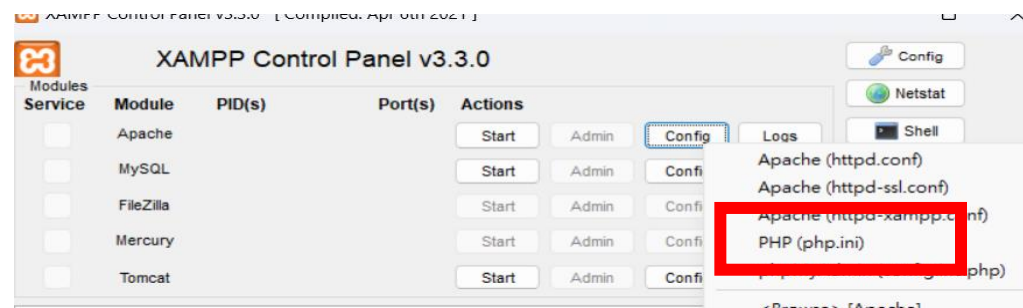
- 於該頁面中，使用**CTRL+F**搜尋**PDO**，檢查是否存在這個擴展，如果存在則可忽略下面的問題。
- 備註：如果不幸找不到PDO擴展的話，可以打開xampp panel，找到apache的config按鈕，選擇php.ini檔案並開啟。
- 找到“`;extension=pdo_mysql`”，將`;`刪除並儲存檔案，重新開啟apache即可。

PDO

PDO support	enabled
PDO drivers	mysql

pdo_mysql

PDO Driver for MySQL	enabled
Client API version	mysqlnd 8.1.6



```
extension=mysql
;extension=oci8_12c ; Use with Oracle Database 12c Instant Client
;extension=oci8_19 ; Use with Oracle Database 19 Instant Client
;extension=odbc
;extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
;extension=pdo_sqlite
```

使用PHP連接資料庫

SDPMLab, NKNU
軟體開發與流程管理實驗室
國立高雄師範大學

使用PHP連接資料庫 – PDO簡介

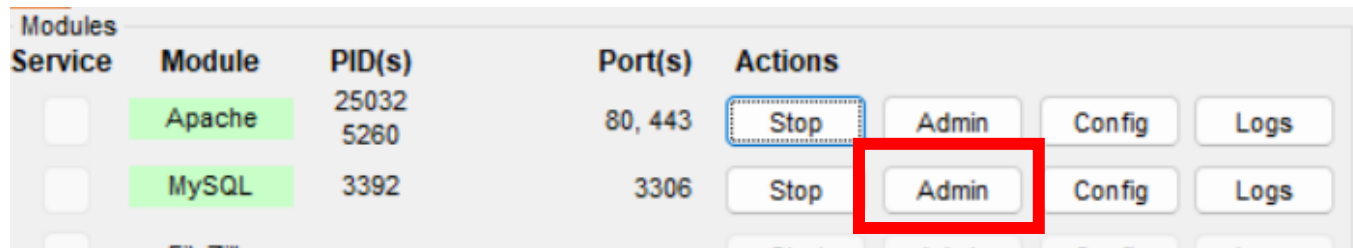
- 該章節主要使用PHP提供的資料庫連線方法**PDO**來實作。
- PDO 全名 **PHP Data Objects**，是一種在PHP裡**連接資料庫的使用介面**，PDO對於資料庫安全性的防範較佳，該方法**可防範 SQL Injection**。



圖片來源:<https://www.cloudways.com/blog/wp-content/uploads/How-to-use-PDO-with-PHP-Banner-1.jpg>

使用PHP連接資料庫 – 建立資料庫

- 點擊XAMPP Panel中MySQL的Admin按鈕，以開啟phpmyadmin。



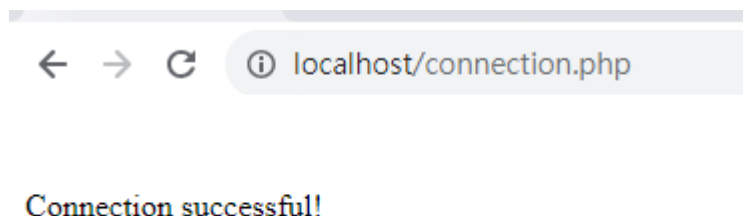
- 點擊左方面板的新增按鈕，並新增資料庫，資料庫名稱依喜好取名，編碼方式請選擇 **utf8_general_ci**。



使用PHP連接資料庫 – 連接資料庫

- 建立一資料夾為Connection，並新增一PHP檔案，名為connection.php.
- 3~6行，為設定**資料庫連接基本參數**。
- 9~11行，為**資料庫連線建立**，以物件方式實例化PDO類別，並將其設定帶入，為PDO實體設定例外屬性。
- 13行，當連線建立失敗時，PHP將會拋出例外訊息。
- 連線成功則會如右下圖所示，出現連線成功訊息。

```
Connection > connection.php > ...
1  <?php
2
3  $serverName = "localhost"; // host,運行於XAMPP時使用localhost即可。
4  $userName   = "root";      // 資料庫之使用者名稱, XAMPP預設root
5  $password   = "";         // 資料庫使用者密碼, XAMPP預設空
6  $db         = "myDB";     // 欲連接的資料庫名稱
7
8  try {
9      $connection = new PDO("mysql:host=$serverName;dbname=$db", $userName, $password);
10     $connection->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11     echo "Connection successful!";
12 } catch(PDOException $e) {
13     echo "Connection failed: " . $e->getMessage();
14 }
15
16 ?>
```



The screenshot shows a web browser window with the address bar displaying 'localhost/connection.php'. Below the address bar, the text 'Connection successful!' is displayed in a green font, indicating that the database connection was successful.

實作 1 連接資料庫

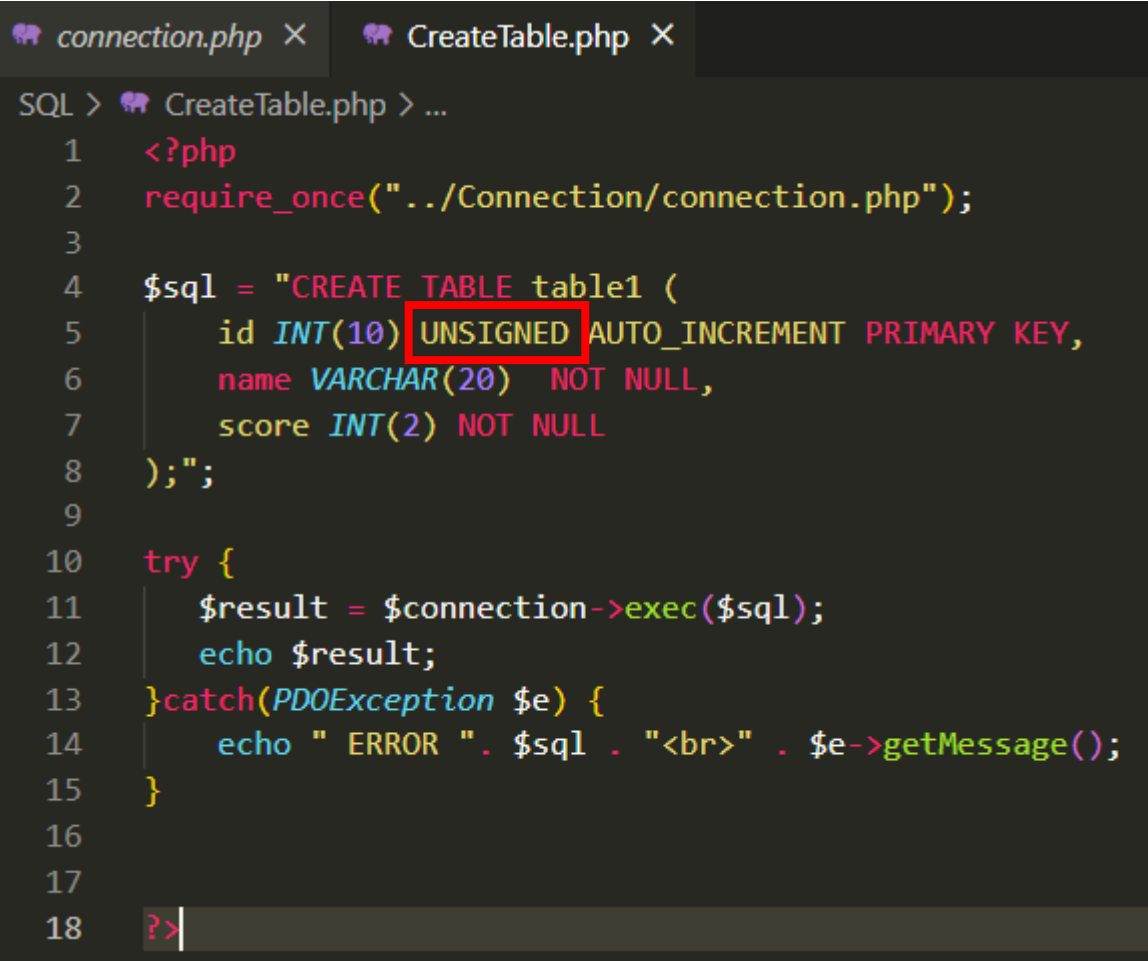
SDPMLab, NKNU
軟體開發與流程管理實驗室
國立高雄師範大學

使用PHP+SQL 進行資料庫操作

SDPMLab, NKNU
軟體開發與流程管理實驗室
國立高雄師範大學

使用PHP+SQL進行資料庫操作- 建立資料表

- 建立一支php檔案，名為CreateTable.php.
- 2行，引入剛剛寫好的資料庫連線方法。
- 4~8行，建立資料表與欄位，**UNSIGNED**將指定欄位為無符號數，即數值限定為0或正數。
- 11行，執行SQL，使用PDO提供的exec方法。
- 結果如下圖即成功。



```
connection.php × CreateTable.php ×
SQL > CreateTable.php > ...
1  <?php
2  require_once("../Connection/connection.php");
3
4  $sql = "CREATE TABLE table1 (
5      id INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
6      name VARCHAR(20) NOT NULL,
7      score INT(2) NOT NULL
8  );";
9
10 try {
11     $result = $connection->exec($sql);
12     echo $result;
13 } catch(PDOException $e) {
14     echo " ERROR ". $sql . "<br>" . $e->getMessage();
15 }
16
17
18 ?>
```

← → ↻ ⓘ localhost/Sample/SQL/CreateTable.php

Connection successful!0

SQL/CreateTable.php

使用PHP+SQL進行資料庫操作- 新增資料

- 建立一支php檔案，名為Insert.php。
- 2行，引入剛剛寫好的資料庫連線方法。
- 4~6行，撰寫新增資料SQL，使用PDO提供的**SQL預處理機制**
將參數抽離(LINE 5)。
- 8~9行，定義將要INSERT的資料。
- 12行，將SQL交給PDO進行預處理。
- 14~16行，將SQL中的 “:NAME” 與 “:SCORE” 進行參數綁定並執行。
- 使用瀏覽器執行該php檔案，即可看到資料庫中多出一筆資料。

	id	name	score
<input type="checkbox"/> 編輯 <input type="checkbox"/> 複製 <input type="checkbox"/> 刪除	1	John	99

```
connection.php CreateTable.php Insert.php X
SQL > Insert.php > ...
1  <?php
2  require_once("../Connection/connection.php");
3
4  $sql = "INSERT INTO table1 (name,score)
5      VALUE (:NAME,:SCORE);
6
7
8  $name = "John";
9  $score = 99;
10
11 try {
12     $prepareSQL = $connection->prepare($sql);
13
14     $prepareSQL->bindParam(":NAME", $name);
15     $prepareSQL->bindParam(":SCORE", $score);
16     $result = $prepareSQL->execute();
17
18     echo $result;
19 } catch(PDOException $e) {
20     echo " ERROR ". $sql . "<br>" . $e->getMessage();
21 }
22
23
24 ?>
```

SQL/Insert.php

實作 2

建立資料表與新增 資料

SDPMLab, NKNU
軟體開發與流程管理實驗室
國立高雄師範大學

使用PHP+SQL進行資料庫操作- 查詢資料

- 於剛剛創建的 myDB 中匯入 **test_table.sql** 以建立資料表。
- 點擊匯入，並上傳sql檔，最終滑到**頁面最下方點擊匯入**即可。
- 資料如下圖所示。

		id	name	math	chinese
<input type="checkbox"/>	編輯 複製 刪除	1	John	99	85
<input type="checkbox"/>	編輯 複製 刪除	2	May	60	100
<input type="checkbox"/>	編輯 複製 刪除	3	Cindy	79	75
<input type="checkbox"/>	編輯 複製 刪除	4	Josh	80	92



匯入到「mydb」資料庫

要匯入的檔案：

可使用壓縮 (gzip, zip) 或無壓縮的檔案。
壓縮檔案名稱必須以 **.[格式].[壓縮方式]** 結尾。如： **.sql.zip**

由電腦上傳：(上限：4,000 MB)

選擇檔案

也可以在任何頁面拖放檔案。

檔案字元編碼：

SQL/test_table.sql

使用PHP+SQL進行資料庫操作- 查詢資料

- 2行，引入資料庫連線方式。
- 4行，**查詢資料的SQL指令**，***** 為搜尋全部資料，遭遇資料量較大的場景下不建議使用。
- 6~8行，執行SQL預處理並執行，使用**setFetchMode()**方法指定資料回傳方式，"**PDO::FETCH_ASSOC**" 為指定回傳須以資料庫中的欄位名稱作為**index**。
- 9~11行，印出資料。

```
1  <?php
2  require_once("../Connection/connection.php");
3
4  $sql = "SELECT * FROM test_table";
5
6  $stmt = $connection->prepare($sql);
7  $stmt->execute();
8  $stmt->setFetchMode(PDO::FETCH_ASSOC);
9  foreach ($stmt->fetchAll() as $key => $value) {
10     echo "id : {$value['id']} , name : {$value['name']} , score : {$value['score']} </br>" ;
11 }
12 ?>
```

← → ↻ ⓘ localhost/Sample//SQL/Select.php

Connection successful!

id : 1 , name : John , math : 99 , chinese : 85
id : 2 , name : May , math : 60 , chinese : 100
id : 3 , name : Cindy , math : 79 , chinese : 75
id : 4 , name : Josh , math : 80 , chinese : 92

SQL/Select.php – lab1 code

實作 3

查詢資料

SDPMLab, NKNU
軟體開發與流程管理實驗室
國立高雄師範大學

實作3 查詢資料

1. 請使用**SELECT**、**FROM**、**WHERE**關鍵字，找出“**John**”的全部分數。
 - 欄位要求：人名、數學成績、國文成績。
2. 請使用**SELECT**、**FROM**、**WHERE**關鍵字搭配**聚合函數**(MAX、COUNT ...)，找出國文最高分(high)、國文最低分(low)、國文總分(sum)及國文平均值(avg)。
 - 備註：使用 **AS** 關鍵字為資料欄位重新命名。
3. 請使用聚合函數**GROUP BY**與**ORDER BY**，將國文分數從多至少排序。

實作3 查詢資料

```
// lab 3 - 1 code
$name = "John";
$sql = "SELECT name,math,chinese FROM test_table WHERE name=:NAME";
$stmt = $connection->prepare($sql);
$stmt->bindParam(':NAME',$name);
$stmt->execute();
$stmt->setFetchMode(PDO::FETCH_ASSOC);
print_r($stmt->fetchAll());
|

// lab 3 - 2 code
$sql = "SELECT MAX(chinese) AS high, MIN(chinese) AS low , SUM(chinese) AS sum ,AVG(chinese) AS avg FROM test_table";
$stmt = $connection->prepare($sql);
$stmt->execute();
$stmt->setFetchMode(PDO::FETCH_ASSOC);
print_r($stmt->fetchAll());

// lab 3 - 3 code
$sql = "SELECT chinese FROM test_table GROUP BY chinese ORDER BY chinese DESC";
$stmt = $connection->prepare($sql);
$stmt->execute();
$stmt->setFetchMode(PDO::FETCH_ASSOC);
print_r($stmt->fetchAll());
```

教育數據分析

SDPMLab, NKTU
軟體開發與流程管理實驗室
國立高雄師範大學