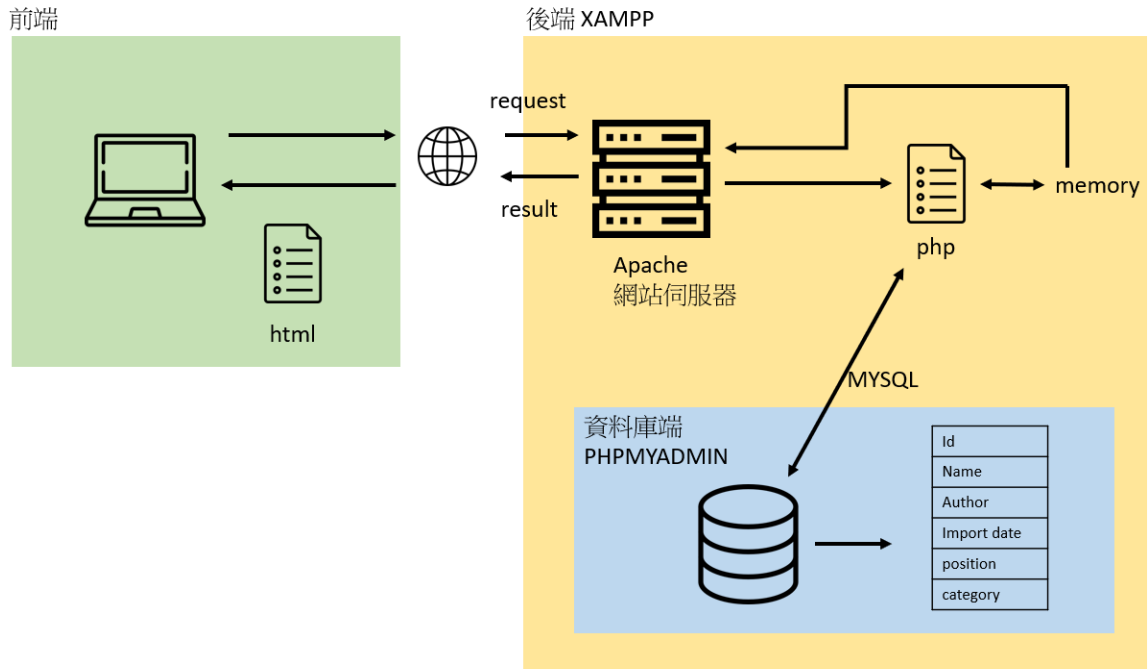


資料庫管理系統 期末專題 報告書

2020 DBMS final project document

姓名：田亦心 學號：P76084075

一、系統架構與環境




主題：圖書館資料庫

作業系統	Windows 10 專業版
前端	HTML
後端	(XAMPP) Apache 網站伺服器 (PHP)
資料庫	(XAMPP) PHPMYADMIN (MYSQL)

二、介面截圖與使用說明

LIBRARY
[Readers Management](#)
[Books Management](#)
[Employees Management](#)
[Borrow Books Management](#)
[Book Venue Management](#)



Reader Management

search reader

reader_id

insert new reader

id

name

register_date(yyyy/mm/dd)

sex

status

delete reader

id

SQL command

array ▼

input your sql command...

- A. 左邊為選取不同頁面之按鈕，包含 reader、book、employee、borrow book、book venue 五種選擇。
- B. 中間為可選取按鈕之頁面，使用者可透過按鈕執行嵌入的 SQL 指令。

search book

book_name ▼

To Kill a Mockingbird

id: 1 - Name: To Kill a Mockingbird -
Author: Harper Lee

insert new book

id

name

author

import_date(yyyy/mm/dd)

position

category

delete book

id

DELETE FROM book WHERE
book_id=11;Record deleted
successfully

New record created successfully

update book

update

id

12

author ▼

rose

submit

Record Update successfully

count book

count

search

13

Employee Management

sum

sum salary

submit

404000

avg

average salary

submit

40400.0000

min

min salary

submit

30000

max

max salary

submit

63000

Borrow Book Management

exists

Who hasn't return books?

submit

id: 1 - Name: Ish Tien
id: 2 - Name: Timmy Wang
id: 5 - Name: Dawn Lee
id: 6 - Name: Alan Hsieh

Who has never borrow books?

submit

id: 7 - Name: Bob Chung
id: 8 - Name: Alex Chen
id: 9 - Name: Vivian Ann
id: 12 - Name: ish

Not exists

Book Venue Management

in

Capacity of meeting Room
IS...

2 ▼

4 ▼

submit

id: 1 - capacity: 2 - hasProjector: 0
id: 2 - capacity: 2 - hasProjector: 0
id: 3 - capacity: 4 - hasProjector: 0
id: 4 - capacity: 4 - hasProjector: 1
id: 5 - capacity: 4 - hasProjector: 1

Capacity of Meeting Room
IS NOT...

2 ▼

4 ▼

submit

id: 6 - capacity: 6 - hasProjector: 0
id: 7 - capacity: 6 - hasProjector: 1
id: 8 - capacity: 8 - hasProjector: 1
id: 9 - capacity: 10 - hasProjector: 1
id: 10 - capacity: 12 - hasProjector: 1

Not in

Who help Readers book venue this year?

having

submit

name: Eric Feng - meeting room id: 2 - book date: 2020-05-27 - isreturned: 1
 name: Ellie Yu - meeting room id: 5 - book date: 2020-05-08 - isreturned: 1
 name: Chloe Hsu - meeting room id: 5 - book date: 2020-05-28 - isreturned: 0
 name: Ben Bai - meeting room id: 7 - book date: 2020-05-01 - isreturned: 1
 name: Ray Peng - meeting room id: 9 - book date: 2020-05-28 - isreturned: 0
 name: Ken Lu - meeting room id: 5 - book date: 2020-05-09 - isreturned: 1
 name: Katie Miao - meeting room id: 6 - book date: 2020-05-28 - isreturned: 0
 name: Betty Kuo - meeting room id: 3 - book date: 2020-05-27 - isreturned: 1

- C. 使用者可透過右邊輸入 SQL 指令，上方的選擇是 SQL 回傳的值，分為 array、integer 與 void。可打入以下指令：

	選單	指令
SELECT-FROM-WHERE	array	SELECT * FROM book WHERE book_name = '1984';
DELETE	no return	DELETE FROM book WHERE book_id= 11;
INSERT	no return	INSERT INTO book (book_id, book_name, author, import_date, position, book_category) VALUES (12, 'ish','amy', '2020/04/02', '000-002-004', 'novel');
UPDATE	no return	UPDATE book SET book_name='sarah' WHERE book_id = 11;
IN	array	SELECT * FROM meeting_room WHERE capacity IN (2, 4) ;
NOT IN	array	SELECT * FROM meeting_room WHERE capacity NOT IN (2, 4) ;
EXISTS	array	SELECT * FROM reader WHERE EXISTS (SELECT * FROM borrow_book WHERE reader.reader_id=borrow_book.reader_id and borrow_book.is_returned = false);
NOT EXISTS	array	SELECT * FROM reader WHERE NOT EXISTS (SELECT * FROM borrow_book WHERE reader.reader_id = borrow_book.reader_id);
COUNT	integer	SELECT COUNT (*) as total FROM book ;
SUM	integer	SELECT SUM (employee_salary) as total FROM

		employee ;
MAX	integer	SELECT MAX (employee_salary) as total FROM employee ;
MIN	integer	SELECT MIN (employee_salary) as total FROM employee ;
AVG	integer	SELECT AVG (employee_salary) as total FROM employee ;
HAVING	array	SELECT employee.employee_id, employee.employee_name, borrow_venue.meeting_room_id, borrow_venue.borrow_date, borrow_venue.is_returned FROM `borrow_venue` JOIN `employee` ON borrow_venue.employee_id = employee.employee_id WHERE borrow_venue.borrow_date BETWEEN '2020-01-01' AND '2020-12-31' GROUP BY employee.employee_id HAVING SUM(borrow_venue.meeting_room_id);

SQL command

array ▼

SELECT * FROM book WHERE
book_name = '1984';

submit

2 1984 George Orwell 2020-05-22
000-000-002 literature

SQL command

no return ▼

INSERT INTO book (book_id,
book_name, author,
import_date, position,
book_category) VALUES (11,
'ish', 'amy', '2020/06/02',
'000-003-004', 'novel');

submit

New record created successfully

SQL command

integer ▼

SELECT COUNT(*) as total
FROM book ;

submit

13

三、資料庫設計

A. 基本要求

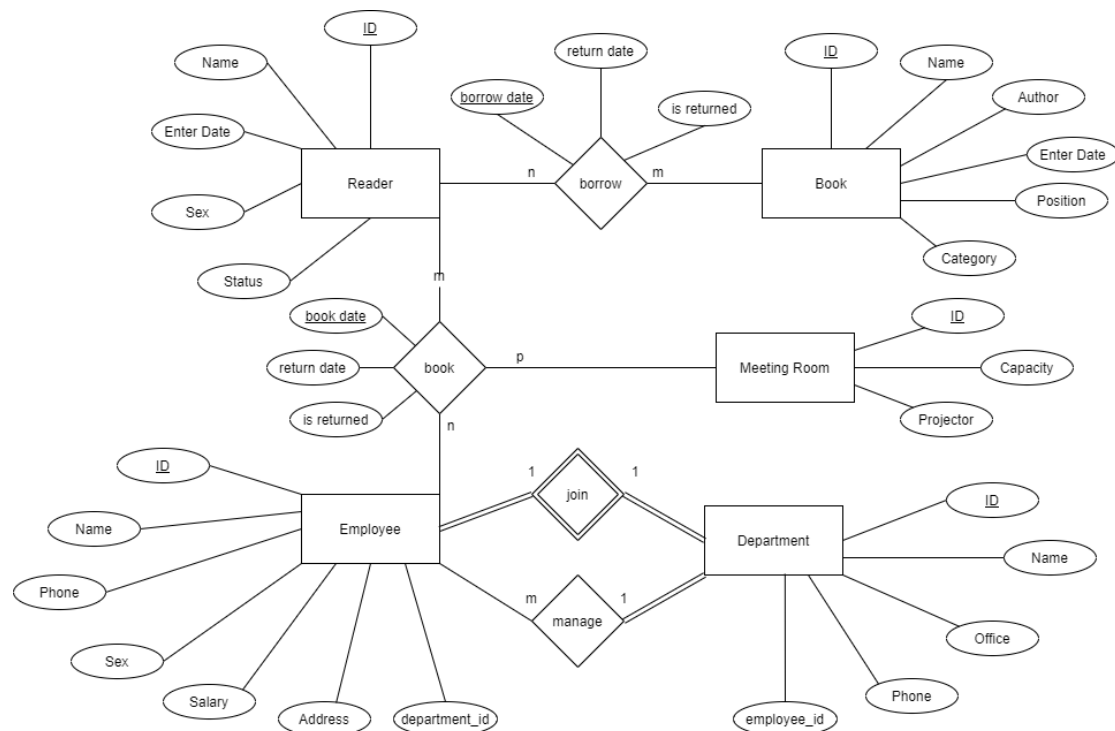
五個 entity: reader、book、meeting room、employee、department

二元關係：join、manage(兩個 entity 間有多個 Relationship)

三元關係：book

Attribute：如圖，每個 entity 都有 id 作為 PK，且都有三個以上的 Attribute

B. ER Model



在圖書館中，有五種 entity，分別是 reader(讀者)、book(書)、meeting room(討論室)、employee(員工)、department(員工的部門)。

C. ER Model Schema

Reader

<u>ID</u>	Name	Enter Date	Sex	Status
-----------	------	------------	-----	--------

讀者(reader)需要 id 為編入的號碼、並且有姓名(name)、入館日期(enter date)以供圖書館在時間到期時註銷讀者、性別(sex)、身分(status)，不同身分如教授、學生權限不同。

Book

<u>ID</u>	Name	Author	Enter Date	Position	Category
-----------	------	--------	------------	----------	----------

書(book)需要 id 為編入的號碼、並且有名字(name)、作者(author)、還有入館日(enter date)以供圖書館參考書的新舊程度、擺放書櫃位置(position)、類別(category)。

Meeting Room

<u>ID</u>	Capacity	Projector
-----------	----------	-----------

討論室(meeting room)有 id 為編入號碼，並包含容納人數(capacity)以及是否有投

影機(projector)。

Employee

<u>ID</u>	Name	Phone	Sex	Salary	Address	<u>Department ID</u>
-----------	------	-------	-----	--------	---------	----------------------

員工(employee)有 id 為編入號碼，還有姓名(name)、電話(phone)、性別(sex)、薪水(salary)、地址(address)。

員工皆須要參加一個部門，同時也被部門管理著，因此，員工有一個外來鍵 department_id。

因為每個員工一定要參加部門，部門也一定要有員工，因此 join 的關係為完全關係。而部門一定要有一個 manager，但員工不一定是 manager，因此 manage 與部門間也是完全關係。

Department

<u>ID</u>	Name	Office	Phone	<u>Manager ID</u>
-----------	------	--------	-------	-------------------

部門(department)有 id 為編入號碼，還有名字(name)、辦公室(office)、電話(phone)。

員工皆須要參加一個部門，同時也被部門管理著，因此，部門有一個外來鍵 manager_id。

因為每個員工一定要參加部門，部門也一定要有員工，因此 join 的關係為完全關係。而部門一定要有一個 manager，但員工不一定是 manager，因此 manage 與部門間也是完全關係。

Borrow Book

<u>Book ID</u>	<u>Reader ID</u>	<u>Borrow Date</u>	Return Date	Is Returned
----------------	------------------	--------------------	-------------	-------------

以下的借閱關係皆為一天。也就是每天一本書只能被一個讀者借閱，一個討論室一天也只能被一個人借。(因此兩個關係中 borrow_date 與 book_date 皆為主鍵的一部份)

讀者可以借很多本書，書也可以被很多人借。因此 borrow 是一個 n 比 m 的關係。並且不是每個人都必須借書、也不是每本書都會被人借。

Book Venue

<u>Reader ID</u>	<u>Employee ID</u>	<u>Meeting Room ID</u>	<u>Borrow Date</u>	Return Date	Is Returned
------------------	--------------------	------------------------	--------------------	-------------	-------------

以下的借閱關係皆為一天。也就是每天一本書只能被一個讀者借閱，一個討論室一天也只能被一個人借。(因此兩個關係中 borrow_date 與 book_date 皆為主鍵的一部份)

討論室也可以被讀者借，但圖書館規定，在借討論室時，必須有一個員工去幫忙開討論室，並且要記錄下來，以防討論室有東西被竊取。一個讀者可以借很多次討論室，並且可以是多個員工幫他借的。員工也可以幫很多讀者借很多不

同討論室。一個討論室也可以被很多讀者使用，被不同員工開門。因此是一個 $n:m:p$ 之關係。但並不是每個讀者都要借討論室、也不是每個員工都必須要去幫忙開門、討論室也不是每間都會被借。