

資料庫期末報告

1. 組員名單

11044106 資管二甲 李欣樺	11044114 資管二甲 辜麗慈
-------------------	-------------------

2. 主題發想

- I. 醫療健保資源是現今台灣引以為傲的全民福利，據統計顯示 110 年度(去年)，台灣每十萬人口中就有高達 92000 的國民有使用到任何門診、住診包含急診等等的醫療資源，透過這高達 90%以上的就診率，指出台灣對於醫療與人民的管理必須更加謹慎且有系統的規劃，使資訊利用能徹底完善，省去不必要的手續及麻煩。
- II. 起初我們希望能做到將傳統 only for 一家醫院的資料庫擴展到 for 多家醫院的雲端資料庫，使民眾能達到互通有無、搜尋篩選最適合自己的就診醫院、科別、醫師，可因為法律、個資法、病患資料...等制式規定，在實際生活中是不可取且無法實現的，因此我們重新聚焦到 for 一家醫院的資料庫，並期望能為其提供多元功能整合使用的資料庫。

3. 希望解決之功能

利用正規化將醫療系統分為四大面向處理

對於醫院	對於人
<ul style="list-style-type: none">✓ 部門與科別 (內科部-肝膽腸胃科)✓ 服務種類 (西醫/中醫/牙醫)	<ul style="list-style-type: none">✓ 醫生基本資料✓ 病患基本資料✓ 優待身份表
對於服務	對於業務
<ul style="list-style-type: none">✓ 症狀查詢與推薦科別✓ 門診表✓ 預約表	<ul style="list-style-type: none">✓ 藥品表/藥品廠商表/藥品明細表✓ 費用項目表/費用項目明細表✓ 掛號費用表✓ 部分負擔金額表 (基本/藥品)

4. 製作過程中的困難

正規化資料庫的過程我們遇到以下難題：

- I. View/Stored Procedure/Function/Cursors 各自該在何種情況下使用？

View	建立一個暫存資料，是資料處理的第一步(可以選擇不用)，可以在 stored procedure 前使用，降低資料抓取錯誤率
Function	需要 return 特定資料並將結果傳給下一個過程處理使用，例如 Stored Procedure 或 Cursors 某一欄位的 value
Stored Procedure	通常為處理資料的最後一步，不需要再將結果拋向下一個地方，只需要查看結果
Cursors	需要遍覽所有資料並對他做處理，通常會再建立一個新 table 將它閱覽的資料插入

- II. Error Code: 1418. This function has none of DETERMINISTIC, NO SQL, or READS SQL DATA in its declaration and binary logging is enabled you *might* want to use the less safe log_bin_trust_function_creators variable

→報錯原因：由於建立自定義 function 時會開啟到

binlog(Binary Logging)，產生資料庫保護機制，而 binlog 是一種二進制日誌記錄，用來記錄資料每一筆轉換的行為(基本上除了 view 與 select 不會變動到資料)

→建立函數時宣告參數表達對資料的處理方式(函數只支援前三個)

DETERMISTIC	確定的，表示在相同的輸入條件下會返回相同的結果，不會隨著時間、隨機性或其他外部因素而變化，進行緩存，提高效率
NO SQL	沒有 SQL 語法（不會修改數據）
READS SQL DATA	只讀取數據（不會修改數據）
MODIFIES SQL DATA	會修改數據
CONTAINS SQL	包含 SQL 語法

III. 資料表與資料表之間該如何做連結？（一對一/一對多/多對多）

→避免多對多的情況，我們會將需要 key in 的 value 再拆分成多個 table 合併表達

Step1：建立一張「症狀對應推薦就診科別表」(多對多)

症狀名稱	推薦科別
頭痛	神經醫學科
頭痛	小兒神經科
頭痛	傳統醫學科
頭痛	精神科
疲倦	肝膽腸胃科
疲倦	新陳代謝科
疲倦	腎臟內科
疲倦	神經醫學科
疲倦	精神科

Step2：將「症狀」與「科別」分別拆成兩張 table

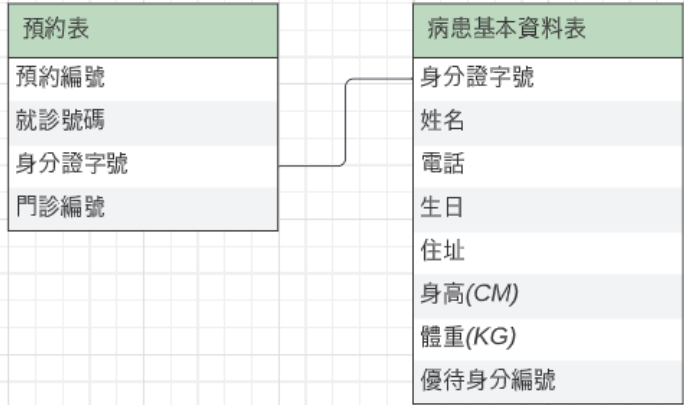
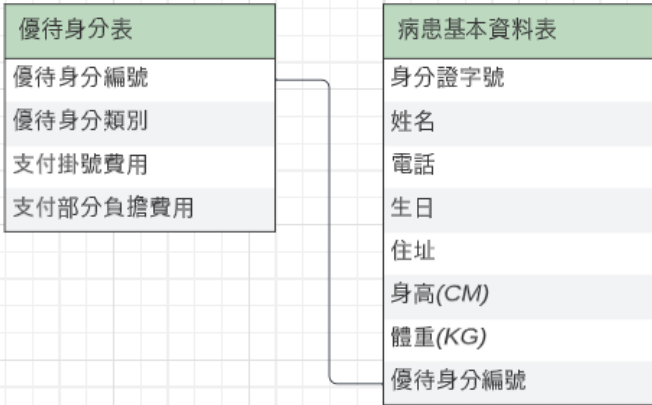
症狀編號	症狀名稱	科別編號	科別名稱
ST0001	頭痛	SS0001	新陳代謝科
ST0002	頭暈	SS0002	肝膽腸胃科
ST0003	嘔吐	SS0003	腎臟內科
ST0004	眼部不適	SS0004	胸腔內科
ST0005	口腔不適	SS0005	過敏免疫風濕科
ST0006	聽力障礙	SS0006	感染防疫科

Step3：再建立一張「症狀推薦科別表」

症狀推薦科別編號	症狀編號	科別編號
SR0001	ST0001	SS0008
SR0002	ST0001	SS0020
SR0003	ST0001	SS0039
SR0004	ST0001	SS0044
SR0005	ST0002	SS0008
SR0006	ST0002	SS0020
SR0007	ST0002	SS0039
SR0008	ST0002	SS0044
SR0009	ST0003	SS0007
SR0010	ST0003	SS0037

IV. 欄位、主鍵、外部鍵的規劃

→避免重複且無必要的欄位名稱、資料與每張資料表的主鍵。外部鍵則是用於確定兩張表的關聯性及資料完整性，規避部分資料匹配不上的問題。而外部鍵在使用時會遇到的問題有以下三點：

種類	問題		
新增資料	子表內有外部鍵，先確認外部鍵連結的父表主鍵欄位資料是否存在 →存在：insert data		
刪除資料	預設模式	當在父表刪除資料時，會比對子表是否有對應的資料，如果有則不允許刪除。	
	ON DELETE 模式	CASCADE	<p>父表刪除資料→子表也刪除</p>  <p>FOREIGN KEY(身分證字號) REFERENCES patient(身分證字號) ON DELETE CASCADE</p>
		SET NULL	<p>父表刪除資料→子表外鍵 NULL/資料不刪除(欄位需允許 NULL)</p>  <p>FOREIGN KEY(優待身分編號) REFERENCES preferential_identity (優待身分編號) ON DELETE SET NULL</p>
編輯資料	編輯和刪除一樣，預設情況下無法直接改變綁定外鍵的欄位 →根據不同需求，在資料表建立時設定對應需求 ON UPDATE 模式		

V. 修改/新增/刪除某個欄位或資料時，是否會造成多個資料表的變動？

→table 需要 key in 不需做計算的 data，相當於死的、不可變動數據，若在有做上述動作時，data 存在於多個 table 或關聯到多個欄位，必定會變更一個以上的數據，為避免此種情況，用「XX 編號」替代名稱變動。

VI. 手動 insert 資料時，對照資料不僅不方便也費時，修改資料也有看錯及改錯疑慮。

→初步想法：資料儲存.csv，再利用 MySQL 提供的 Table Data Import Wizard 把 csv 直接匯入 MySQL 內。

→最終方案：(規定只能用 insert 新增資料)

運用 Python 將 csv 資料依照特定 table 轉成 MySQL insert

```
1 # 引入 pandas 模組並取名為 pd
2 import pandas as pd
3
4 # csv檔案路徑
5 file_path = '掛號費用表.csv'
6
7 # insert的資料表名稱
8 table = 'registration_fee'
9
10 # 讀取csv
11 df = pd.read_csv(file_path)
12
13 # 把csv內資料依照特定table轉成MySQL的insert語法
14 rowLen = len(df)
15 colLen = len(list(df))
16 for i in range(rowLen):
17     print("INSERT INTO "+table+" VALUES (", end="")
18     for j in range(colLen):
19         if j != colLen-1:
20             if type(df.iat[i, j]) == str:
21                 print("'" + str(df.iat[i, j]) + "'", end="")
22             else:
23                 print(str(df.iat[i, j]) + ",", end="")
24         else:
25             if type(df.iat[i, j]) == str:
26                 print("'" + str(df.iat[i, j]) + "'", end="")
27             else:
28                 print(str(df.iat[i, j]), end="")
29     print(");")
```

執行結果：

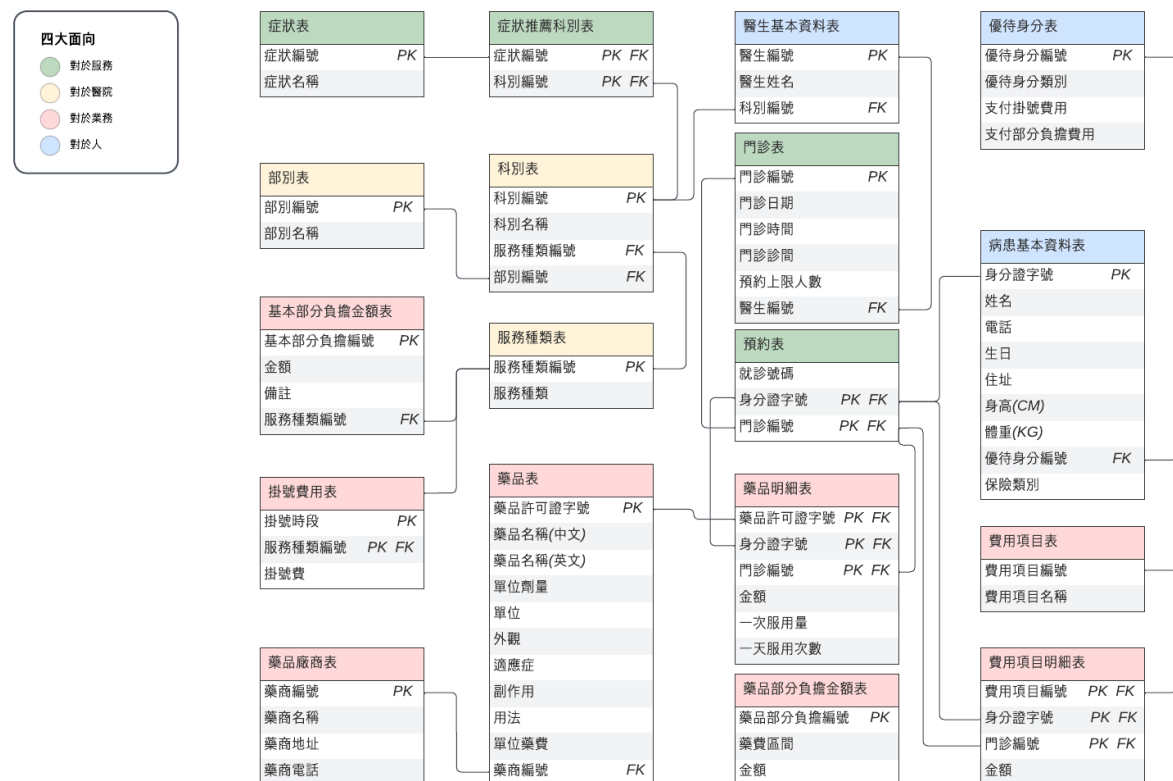
```
PS E:\1112課程\資料庫管理\醫院系統\csv> python -u "e:\1112課程\資料庫管理\醫院系統\csv\csvToMySQL_insert.py"
INSERT INTO registration_fee VALUES ('RF0001','夜間門診',200,'SC0001');
INSERT INTO registration_fee VALUES ('RF0002','週六門診',200,'SC0001');
INSERT INTO registration_fee VALUES ('RF0003','其它時段',150,'SC0001');
INSERT INTO registration_fee VALUES ('RF0004','夜間門診',200,'SC0002');
INSERT INTO registration_fee VALUES ('RF0005','週六門診',200,'SC0002');
INSERT INTO registration_fee VALUES ('RF0006','其它時段',150,'SC0002');
INSERT INTO registration_fee VALUES ('RF0007','夜間門診',200,'SC0003');
INSERT INTO registration_fee VALUES ('RF0008','週六門診',200,'SC0003');
INSERT INTO registration_fee VALUES ('RF0009','其它時段',150,'SC0003');
```

5. 設計過程與展示

正規化分為 18 張資料表：

門診表	醫生基本資料表	部別表
科別表	服務種類表	症狀表
症狀推薦科別表	病患基本資料表	優待身分表
預約表	費用項目表	費用項目明細表
基本部分負擔金額表	藥品部分負擔金額表	掛號費用表
藥品表	藥品廠商表	藥品明細表

醫療系統架構圖



展示功能依據兩種不同使用者角度展示：

I. 病人操作 (以病人身份 N123450011)

有一個人叫林一一(N123450011)，她已經頭痛好幾天了，但死都不篩，因為她覺得光是頭痛這個症狀不代表又二確，因此為了確定還有什麼原因造成頭痛，就運用醫院的查詢功能能看哪一科，最後選擇神經醫學科很有名的方樂昕醫生，也順便預約可以前往的門診與時間。只是到當天她就忘記自己的就診號碼，因此又上網站查詢發現自己是一號，趕忙出門的他心想：「幸好有這個功能，不然就過號了」。看完診後，醫生說可能只是上次確診留下的後遺症，叫她回去少打一點 code，多休息就好。最後林一一就拿著藥事單與批價單快樂繳錢領藥回家了。
P.S.大家去醫院記得戴口罩唷！

A. 查詢頭痛症狀屬於哪些科別

→運用 stored procedure 輸入症狀名稱即可查詢有什麼科別

```
delimiter $$
create procedure symptom_for_sessions(symptom_name varchar(10))
begin
    SELECT symptom.症狀名稱, sessions.科別名稱
    FROM symptom_recommend
    INNER JOIN sessions ON symptom_recommend.科別編號 = sessions.科別編號
    INNER JOIN symptom ON symptom_recommend.症狀編號 = symptom.症狀編號
    WHERE symptom.症狀名稱 = symptom_name;
end$$
delimiter ;

call symptom_for_sessions("頭痛");
```

B. 選擇神經醫學科，查詢神經醫學科的所有醫生

→運用 stored procedure 輸入科別名稱即可查詢有什麼醫生

```

delimiter $$
create procedure doctors_in_sessions(sessions_name varchar(20))
) begin
    SELECT sessions.科別名稱, doctor.醫生姓名
    FROM doctor
    INNER JOIN sessions ON doctor.科別編號 = sessions.科別編號
    WHERE sessions.科別名稱 = sessions_name;
end$$
delimiter ;

call doctors_in_sessions("神經醫學科");

```

C. 選擇方樂昕醫生，查詢方樂昕醫生的門診時間

→運用 stored procedure 輸入醫生名稱即可查詢有什麼門診

```

delimiter $$
create procedure doctors_clinic(doctors_name varchar(4))
begin
    SELECT doctor.醫生姓名, sessions.科別名稱, clinic.門診日期, clinic.門診時間, clinic.門診診間, clinic.預約上限人數
    FROM clinic
    INNER JOIN doctor ON clinic.醫生編號 = doctor.醫生編號
    INNER JOIN sessions ON doctor.科別編號 = sessions.科別編號
    WHERE doctor.醫生姓名 = doctors_name;
end$$
delimiter ;

call doctors_clinic("方樂昕");

```

D. 確定預約，查詢身分證字號為 N123450011 的姓名、預約門診、就診號碼

→運用 stored procedure 輸入身分證字號即可查詢有什麼就診號碼與預約資料

```

delimiter $$
create procedure patient_appointment(patientID varchar(10))
begin
    SELECT patient.姓名, doctor.醫生姓名, sessions.科別名稱, appointment.就診號碼, clinic.門診診間, clinic.門診日期, clinic.門診時間
    FROM patient
    INNER JOIN appointment ON appointment.身分證字號 = patient.身分證字號
    INNER JOIN clinic ON clinic.門診編號 = appointment.門診編號
    INNER JOIN doctor ON doctor.醫生編號 = clinic.醫生編號
    INNER JOIN sessions ON sessions.科別編號 = doctor.科別編號
    WHERE patient.身分證字號 = patientID;
end$$
delimiter ;

call patient_appointment("N123450011");

```


E. 看診完畢，醫生開立的藥事單

→建立 view 先暫存處方籤會使用到的內容

```
drop view if exists prescriptions;
create view prescriptions as
select medicine_detail.身分證字號, medicine_detail.門診編號, medicine.`藥品名稱(英文)`, medicine_detail.一次服用量,
medicine_detail.一天服用次數, (medicine_detail.一次服用量 * medicine_detail.一天服用次數) as 總量,
(medicine_detail.一次服用量 * medicine_detail.一天服用次數 * medicine.單位藥費) as 每種藥品藥費
from medicine_detail
join medicine on medicine_detail.藥品許可證字號 = medicine.藥品許可證字號;
```

→建立 stored procedure 輸入身分證字號與門診編號抓到這位病人今天看診所拿到的所有藥品

```
DELIMITER $$
DROP PROCEDURE IF EXISTS prescriptions_for_clinic; $$
CREATE PROCEDURE prescriptions_for_clinic (IN 身分證字號 varchar(10), IN 門診編號 VARCHAR(6))
BEGIN

    select prescriptions.`藥品名稱(英文)`, prescriptions.一次服用量,
    prescriptions.一天服用次數, prescriptions.總量,
    prescriptions.每種藥品藥費
    from prescriptions
    where 身分證字號 = prescriptions.身分證字號
    and 門診編號=prescriptions.門診編號;

END $$
DELIMITER ;

call prescriptions_for_clinic ('N123450011', 'CN0029');
```

→建立 function 計算所有藥品總共的花費(輸入身分證字號)

```
delimiter $$
create function total_medicine_expense(patientID varchar(10))
returns int
DETERMINISTIC
begin
    declare total_medicine_expense int;
    select sum(每種藥品藥費) into total_medicine_expense
    from prescriptions
    where 身分證字號 = patientID;
    return total_medicine_expense;
end$$
delimiter ;
```

→根據藥品總花費利用 funtion+case...when 判斷總藥費級距
以此分類藥品部分負擔金額

```

delimiter $$
create function medicine_expense_sort(expense int)
returns int
DETERMINISTIC
begin
    declare medicine_partial int;
    case
    when expense between 0 and 100 then set medicine_partial=0;
    when expense between 101 and 200 then set medicine_partial=20;
    when expense between 201 and 300 then set medicine_partial=40;
    when expense between 301 and 400 then set medicine_partial=60;
    when expense between 401 and 500 then set medicine_partial=80;
    when expense between 501 and 600 then set medicine_partial=100;
    end case;
    return medicine_partial;
end$$
delimiter ;

```

F. 看診完畢，拿到醫療費用收據

→建立 view 判斷各類費用對於不同保險類別的人屬於健保或自費，以及健保點數或自費金額應如何計算。

```

drop view if exists fee_receipt;
create view fee_receipt as
SELECT
patient.姓名,
patient.身分證字號,
clinic.門診編號,
clinic.門診日期,
clinic.門診時間,
sessions.科別名稱,
expense_items.費用項目名稱,
CASE
WHEN patient.保險類別 = '健保'
AND (expense_items.費用項目名稱 = '檢驗檢查費' OR expense_items.費用項目名稱 = '藥品費' OR expense_items.費用項目名稱 = '證明書費'
OR expense_items.費用項目名稱 = '差額負擔' OR expense_items.費用項目名稱 = '其他項目' OR expense_items.費用項目名稱 = '衛材費')
THEN '自費金額'
WHEN patient.保險類別 = '自費'
THEN '自費金額'
ELSE '健保申報點數' END AS `健保/自費`,
expense_items_detail.金額 as 費用
FROM clinic,sessions,doctor,expense_items,patient,expense_items_detail
WHERE (clinic.門診編號=expense_items_detail.門診編號)
AND (sessions.科別編號=doctor.科別編號)
AND (doctor.醫生編號=clinic.醫生編號)
AND (expense_items.費用項目編號=expense_items_detail.費用項目編號)
AND (patient.身分證字號=expense_items_detail.身分證字號)
ORDER BY `健保/自費`;

```

→建立 stored procedure 輸入身分證字號及門診編號查詢當次看診的部分費用明細

```

DELIMITER $$
DROP PROCEDURE IF EXISTS patient_fee_receipt; $$
CREATE PROCEDURE patient_fee_receipt (IN 身分證字號 varchar(10),IN 門診編號 VARCHAR(6))
BEGIN

    select 費用項目名稱, `健保/自費`, 費用
    from fee_receipt
    where 身分證字號=fee_receipt.身分證字號
    and 門診編號=fee_receipt.門診編號;

END $$
DELIMITER ;

CALL patient_fee_receipt('N123450011','CN0029');

```

→建立 function 判斷部分費用項目的總健保申報點數(後面建立完整的費用收據會使用到此 function)

```

delimiter $$
drop function if exists total_health_insurance_point; $$
create function total_health_insurance_point(身分證字號 varchar(10),門診編號 varchar(6))
returns int
DETERMINISTIC
begin
    declare 總健保申報點數 int;
    select sum(fee_receipt.費用)
    as 總健保申報點數
    into 總健保申報點數
    from fee_receipt
    where fee_receipt.`健保/自費`='健保申報點數'
    and 門診編號=fee_receipt.門診編號;
    return 總健保申報點數;
end$$
delimiter ;

```

→建立 function 判斷部分費用項目的總自費金額(後面建立完整的費用收據會使用到此 function)

```

delimiter $$
drop function if exists total_self_pay; $$
create function total_self_pay(身分證字號 varchar(10),門診編號 varchar(6))
returns int
DETERMINISTIC
begin
    declare 總自費金額 int;
    select sum(fee_receipt.費用)
    as 總自費金額
    into 總自費金額
    from fee_receipt
    where fee_receipt.`健保/自費`='自費金額'
    and 門診編號=fee_receipt.門診編號;
    return 總自費金額;
end$$
delimiter ;

```

→建立 function 判斷門診屬於哪個掛號時段(後面建立完整的費用收據會使用到此 function)

```

delimiter $$
drop function if exists clinic_registration_time; $$
create function clinic_registration_time(門診編號 varchar(6))
returns varchar(4)
DETERMINISTIC
begin
    declare 掛號時段 varchar(4);
    select
    case
    when DATE_FORMAT(clinic.門診日期, '%w')=6 then '週六門診'
    when clinic.門診時間='晚間' then '夜間門診'
    else '其他門診' end
    as 掛號時段
    into 掛號時段
    from clinic
    where 門診編號=clinic.門診編號;
    return 掛號時段;
end$$
delimiter ;

```

→運用 stored procedure 輸入身分證字號、門診編號、(經轉診/未經轉診)，可查詢當次看診的費用收據

```

DELIMITER $$
DROP PROCEDURE IF EXISTS patient_fee_information; $$
CREATE PROCEDURE patient_fee_information (IN 身分證字號 varchar(10),IN 門診編號 VARCHAR(6),IN 備註 varchar(20))
BEGIN
    select distinct 姓名,身分證字號,門診編號,門診日期,門診時間,fee_receipt.科別名稱,
    total_medicine_expense("N123450011") as 藥費,total_health_insurance_point(身分證字號,門診編號) as 健保申報點數,
    (total_medicine_expense("N123450011")+total_health_insurance_point(身分證字號,門診編號)) AS 合計健保申報點數,
    registration_fee,掛號費,basic_partial_burden.金額 as 基本部分負擔金額,
    medicine_expense_sort(total_medicine_expense("N123450011")) as 藥品部分負擔金額,
    total_self_pay(身分證字號,門診編號) as 其他自費金額,
    (registration_fee,掛號費+basic_partial_burden.金額+medicine_expense_sort(total_medicine_expense("N123450011"))+total_self_pay(身分證字號,門診編號)) as 應繳金額
    from sessions
    join fee_receipt on sessions.科別名稱 = fee_receipt.科別名稱
    join service_class on sessions.服務種類編號=service_class.服務種類編號
    join registration_fee on service_class.服務種類編號=registration_fee.服務種類編號
    join basic_partial_burden on service_class.服務種類編號=basic_partial_burden.服務種類編號
    where 身分證字號=fee_receipt.身分證字號
    and 門診編號=fee_receipt.門診編號
    and 備註=basic_partial_burden.備註
    and clinic_registration_time(門診編號)=registration_fee.掛號時段;
END $$
DELIMITER ;

CALL patient_fee_information('N123450011','CN0029','經轉診');

```

II. 醫生操作 (以方樂昕醫生為例)

社畜的一天又開始了，方樂昕醫生到醫院後運用醫生身份登入後台查看今天又有多少門診要看，順便大致了解今天都有哪些病人、有沒有需要注意的地方。而看到今天自己有個門診只有兩個人就可以下班耍廢，都不知道是要高興還是難過了QQ

A. 目前方樂昕醫生門診預約人數

→建立 view 暫存所有門診預約人數

```

create view appointment_trial as
select 門診編號, count(*) as 目前門診預約人數
from appointment
group by 門診編號
order by 門診編號 ASC;
select * from appointment_trial;

```

→建立 stored procedure 輸入醫生姓名即可得到此人門診編號、目前預約人數、以及運用 case...when 判斷是否可以現場掛號 (目前預約人數 < 預約上限人數)

```

delimiter $$
create procedure appointment_number(doctor_name varchar(4))
begin
    select appointment_trial.門診編號, appointment_trial.目前門診預約人數, doctor.醫生姓名,
    case
    when appointment_trial.目前門診預約人數 > clinic.預約上限人數 then "否"
    else "是" end as 是否可以現場掛號
    from appointment_trial
    join clinic on appointment_trial.門診編號 = clinic.門診編號
    join doctor on clinic.醫生編號 = doctor.醫生編號
    where doctor.醫生姓名 = doctor_name;
end$$
delimiter ;

call appointment_number("方樂昕");

```

B. 門診 CN0029 詳細資料

→建立 stored procedure 輸入門診編號即可得到預約此門診的病人簡述資料

```

delimiter $$
create procedure appointment_detail(clinicID varchar(6))
begin
    select appointment.就診號碼, appointment.門診編號, appointment.身分證字號, patient.姓名
    from appointment
    join patient on patient.身分證字號 = appointment.身分證字號
    where appointment.門診編號 = clinicID
    order by appointment.就診號碼;
end$$
delimiter ;

call appointment_detail("CN0029");

```

6. 分工與心得

11044106 資管二甲 李欣樺	11044114 資管二甲 辜麗慈
報告 PPT 檔	報告書面檔
MySQL	MySQL
口頭報告	口頭報告

- ✓ 李欣樺：我覺得這次專題遇到的困難有兩點。第一，設計資料庫時，因為要符合真實性與完整性，但醫療系統內部的資料有些是不公開的，我們上網查找相關資料是有點難找到的，因此我們在

蒐集所有可能的資料欄位與理解這些資料的運作邏輯就需要花很多時間，在建資料庫也會需要不斷進行微調，來讓此資料庫更加真實與完整。第二，在實現批價功能時，我們需要抓去接近 10 個資料表去進行判斷與計算，最後整理成符合現實醫療系統的功能表。在這當中，我們用到了 view,stored procedure,stored function,cursor 等 sql 的語法，可以說是把期末所有教到的內容都實作了一次。從中我會思考每種狀況分別使用哪種方法去實現，什麼樣的順序與邏輯才是順的、可行的。很多時候，我都是在思考，反而不是純寫 code，因為我必須先理清邏輯，先想過什麼樣才是好的設計，才不會在建立資料庫後，發現結構有很大的問題，導致實現不了查詢、新增、修改、刪除的功能。設計好的結構的資料庫，是我認為這門課最重要的能力，且必須嚴謹看待。

- ✓ 辜麗慈：經過這學期資料庫的學習，我認為收穫最大的就是這份專題報告，即使透過平時上課的批批踢與老師的講述，我們也只能明白每一個功能、每一個語法的理論知識，實際操作的經驗也要等到真的用到才知道，例如 function 的使用由於老師上課是建立一個不需要讀取資料的自定義函數，我們只需要輸入參數就能運作就好，但實際運用在操作資料時就會出現錯誤訊息，由於資料庫的保護機制，以及不個關鍵字的權限與設計都是有其意義的。而在設計專題的前期最難的應該是正規化，在一開始我們一籬筐把我們想到的所有功能、欄位、想要實現的目標全部攤開來，需要一樣一樣分門別類之外，由於不是所有東西都有明確的劃分界線，根據不同的角度看待會有不同的分類方向，而這些定義衝突導致我們在在初期會一直刪刪減減並懷疑自己的決定，幸

好當時有在諮詢時間與老師商量，從老師那裡得知資料庫是需要符合真實性與完整性，並且引導我們應該先從主題下手，再參考真實的物件，接著列出功能表，依據功能把相應的資料表生出來。我認為這很好的訓練我處理事情並聚焦問題的能力，讓我不會在下次面對磚石時又先處理或幻想中後期才會遇到的邏輯與難題，應該腳踏實地一步一步來。後期則是大觀念的建立，view/function/stored procedure/cursor 的選擇是一大難題，基本上的操作都學會，但我在期末完善資料庫時，還是盯著這些東西想很久，想著該如何「恰當」地使用它們，最後得出的結果也打在書面資料中，我認為這學期資料庫與專題的學習受益匪淺，只剩下 Azure 還等著我去破關了！

7. 報告提問的回覆

- a) E-R Model 希望能在能標一對一跟多對多
- b) Q：在查詢門診資料表加上目前預約人數有多少 讓病人判斷能否預約？ A：但在 table 裡是不能建立此欄位將計算值傳回，應該是在 view 裡建立，那後面是有這項功能的

```
821 delimiter $$
822 • create procedure appointment_number(doctor_name varchar(4))
823 begin
824     select appointment_trial.門診編號, appointment_trial.目前門診預約人數, doctor.醫生姓名,
825     case
826     when appointment_trial.目前門診預約人數 > clinic.預約上限人數 then "否"
827     else "是" end as 是否可以現場掛號
828     from appointment_trial
829     join clinic on appointment_trial.門診編號 = clinic.門診編號
830     join doctor on clinic.醫生編號 = doctor.醫生編號
831     where doctor.醫生姓名 = doctor_name;
832 end$$
833 delimiter ;
834
835 • call appointment_number("方樂昕");
836
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

門診編號	目前門診預約人數	醫生姓名	是否可以現場掛號
CN0029	2	方樂昕	是