

系統說明文件

1. 系統架構與環境

用 python 與 MySQL Workbench 實作此專題，利用 python 自己的套件 `mysql.connector` 與此資料庫做連接，也用 `tkinter` 套件來寫介面。

2. 介面截圖與使用說明

The screenshot shows the main interface of the DBMS Project application. It features a central panel with multiple sections for database operations, each with input fields and a corresponding button:

- Select from where:** Includes input fields for 'Select', 'from', and 'where', with a 'SELECT FROM WHERE' button.
- delete table, delete attribute, delete data:** Includes input fields and a yellow 'Choose To Delete' button.
- Insert table, Insert data:** Includes input fields and a red 'INSERT' button.
- update table, update schema, update data, update key attribute, update key data:** Includes input fields and a blue 'UPDATE' button.
- condition condition:** Includes input fields and buttons for 'IN' (orange), 'NOT IN' (purple), and 'EXISTS' (pink).
- count (), count table, condition:** Includes input fields and a yellow 'COUNT' button.
- sum (), count table, condition:** Includes input fields and a red 'SUM' button.
- max (), max table, condition:** Includes input fields and a blue 'MAX' button.
- min (), min table, condition:** Includes input fields and an orange 'MIN' button.
- avg (), avg table, condition:** Includes input fields and a purple 'AVG' button.
- select, hav table, where condition, group by, having condition:** Includes input fields and a pink 'HAV' button.
- Query:** Includes a text input field and a grey 'QUERY' button.

總介面如右圖

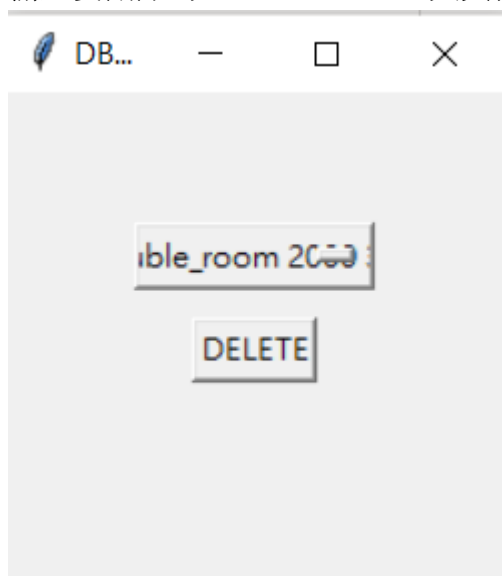
This screenshot shows a specific part of the DBMS Project interface. On the left, the 'Select from where' section is active, with input fields for 'Select', 'from', and 'where', and a 'SELECT FROM WHERE' button. On the right, a list of room attendant names is displayed:

- 300 Jack room_attendant
- 301 Bob receptionist
- 302 Rose room_attendant
- 303 Que cook
- 304 Poo receptionist
- 305 Lion room_attendant
- 306 Tiger cook
- 307 Weed receptionist
- 308 Fish room_attendant
- 309 Zhong room_attendant

輸入 select 的條件，如想輸出的 attribute。輸入要使用指令的 table。輸入條件，如某 `e_id < 310`。右圖為輸出結果。

This screenshot shows the 'delete table, delete attribute, delete data' section of the DBMS Project interface. It includes input fields for each operation and a yellow 'Choose To Delete' button.

輸入要刪除的 table、attribute 與其值。其餘依此類推。



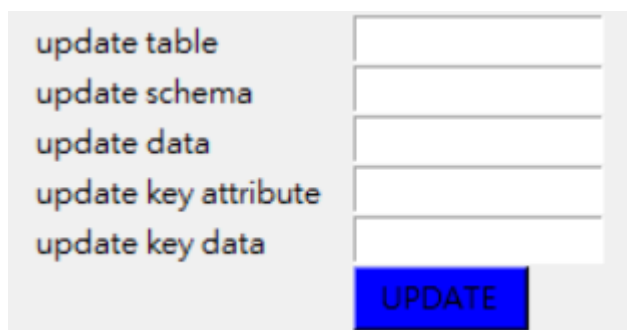
A screenshot of a database window titled "DB...". Inside the window, there is a text input field containing the text "table_room 2000". Below the input field is a button labeled "DELETE".

並會提供一個下拉是選單去選符合條件的去刪除。



A screenshot of a database window showing two labels: "Insert table" and "Insert data". Below each label is an empty text input field. At the bottom right of the window is a red button labeled "INSERT".

選擇要 insert 的 table 如 room。要 insert 的 data 如 400,'double_room',2000



A screenshot of a database window showing a list of update options: "update table", "update schema", "update data", "update key attribute", and "update key data". To the right of these options are five empty text input fields. At the bottom right of the window is a blue button labeled "UPDATE".

寫入要更新的 table 如 room。要更新的 schema 如 type='single_room'。要更新的 data 如 'double_room'。該 table 的 key attribute 與其值如 r_id 與 400。

condition	<input type="text"/>	IN
condition	<input type="text"/>	
condition	<input type="text"/>	NOT IN
condition	<input type="text"/>	
condition	<input type="text"/>	EXISTS
condition	<input type="text"/>	
condition	<input type="text"/>	NOT EXISTS
condition	<input type="text"/>	

由於上圖皆為 **nested query operator**，所以輸入皆會以 **sql** 的形式做輸入，再直接使用按鈕上的該 **function**。

count ()	<input type="text"/>	COUNT
count table	<input type="text"/>	
condition	<input type="text"/>	
sum ()	<input type="text"/>	SUM
count table	<input type="text"/>	
condition	<input type="text"/>	
max ()	<input type="text"/>	MAX
max table	<input type="text"/>	
condition	<input type="text"/>	
min ()	<input type="text"/>	MIN
min table	<input type="text"/>	
condition	<input type="text"/>	
avg ()	<input type="text"/>	AVG
avg table	<input type="text"/>	
condition	<input type="text"/>	

上圖的輸入方式皆相同。一開始皆為輸入要記數的、加總的、找最大值的、找最小值的、算平均值的 **attribute**。接著輸入該 **attribute** 的 **table**。最後是查詢條件。

select	<input type="text"/>
hav table	<input type="text"/>
where condition	<input type="text"/>
group by	<input type="text"/>
having condition	<input type="text"/>
	HAV

上圖為 **having** 的實作。輸入要 **select** 取出的值。再輸入相應的 **table**。輸入

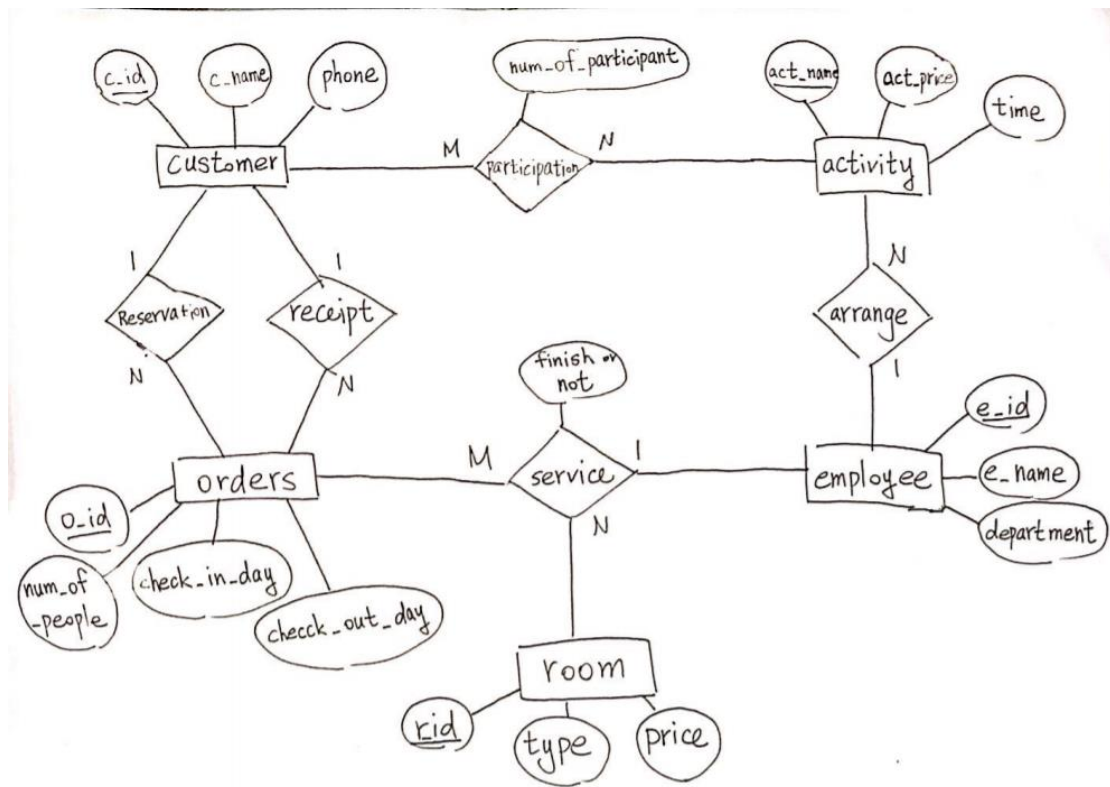
對該 table 的條件。再輸入要將哪個 attribute 相同的做合併。最後在寫入合併後的條件。

Query

QUERY

此為 Query 的輸入方式。只需要輸入 sql 指令即可。

3. ER diagram



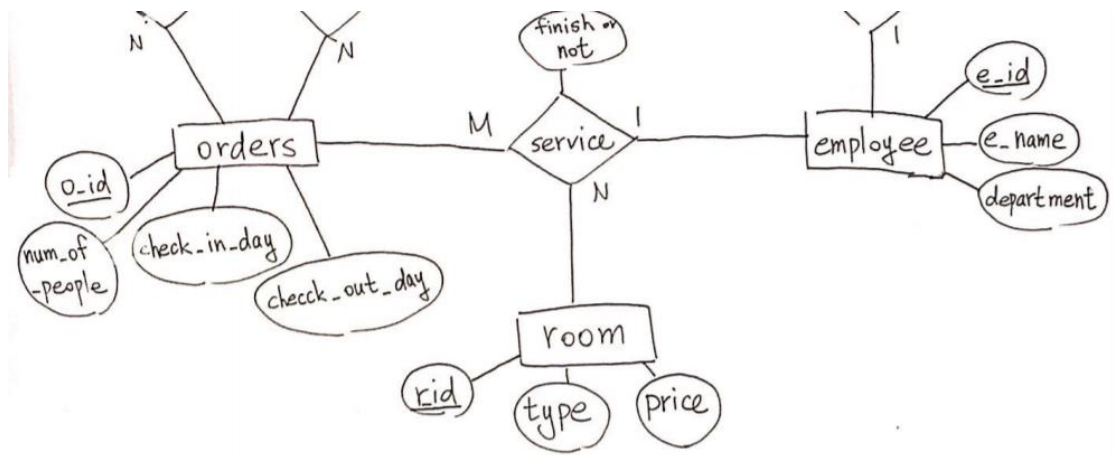
4. 第三正規化後的 Relation Schema

customer (c-id , c_name , phone)
orders (o-id , check-in-day , check-out-day , num-of-people ,
c-id , e-id)
activity (act-name , act-price , time , e-id)
participation (c-id , act-name , num-of-participant)
employee (e-id , e_name , department)
room (r-id , type , price , e-id)
service (o-id , r-id , finish-or-not)

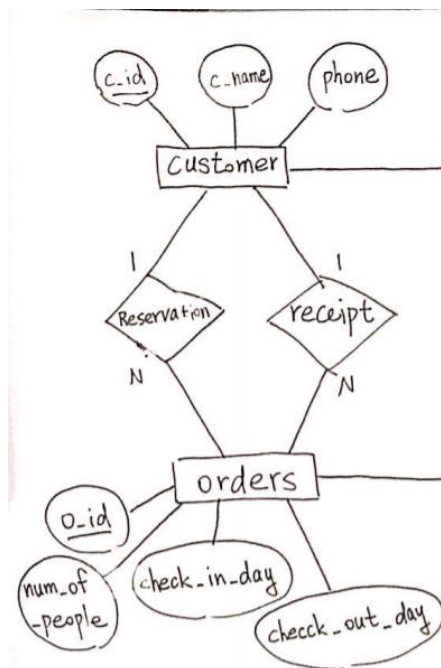
5. 規定數量

有 5 個 entity : customer, activity, orders, room, employee

皆有二元與三元關係



也有多個 relationship



每個 table 都有 3 個 attributes，每個 table 皆有 key attribute，每個 table 皆有 10 筆不同的資料，如下圖：

	c_id	c_name	phone	o_id	r_id	finish_or_not
▶	100	Steve	0925068800	200	402	yes
	101	Max	0925068801	201	407	no
	102	Chang	0925068802	202	409	yes
	103	Wilson	0925068803	203	401	no
	104	Wei	0925068804	204	405	yes
	105	Ping	0925068805	205	400	yes
	106	Yue	0925068806	206	404	no
	107	Hang	0925068807	207	403	yes
	108	Zheng	0925068808	208	406	yes
	109	Lung	0925068809	209	408	no
*	NULL	NULL	NULL	NULL	NULL	NULL

	act_name	act_price	time	e_id	c_id	act_name	num_of_participant
▶	film	200	20210509	305	100	swimming	3
	fishing	400	20210510	308	101	kayaking	3
	karaoke	600	20210510	307	102	spa	3
	kayaking	1300	20210507	303	103	snorkeling	2
	massage	800	20210504	300	104	massage	1
	scuba_diving	2200	20210510	306	105	workout	5
	snorkeling	500	20210505	301	106	film	2
	spa	200	20210506	302	107	scuba_diving	2
	swimming	500	20210515	309	108	fishing	2
	workout	250	20210508	304	109	karaoke	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	e_id	e_name	department	r_id	type	price	e_id
▶	300	Jack	room_attendant	400	double_room	2000	305
	301	Bob	receptionist	401	single_room	1000	303
	302	Rose	room_attendant	402	single_room	1000	300
	303	Que	cook	403	family_room	5000	307
	304	Poo	receptionist	404	single_room	1000	306
	305	Lion	room_attendant	405	family_room	5000	304
	306	Tiger	cook	406	family_room	5000	308
	307	Weed	receptionist	407	double_room	2000	301
	308	Fish	room_attendant	408	double_room	2000	309
	309	Zhong	room_attendant	409	family_room	5000	302
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	o_id	check_in_day	check_out_day	num_of_people	c_id	e_id
▶	200	20210501	20210504	5	100	300
	201	20210502	20210505	4	101	301
	202	20210503	20210506	3	102	302
	203	20210504	20210507	2	103	303
	204	20210505	20210508	1	104	304
	205	20210506	20210509	9	105	305
	206	20210507	20210510	2	106	306
	207	20210508	20210510	4	107	307
	208	20210509	20210510	2	108	308
	209	20210510	20210515	5	109	309
*	NULL	NULL	NULL	NULL	NULL	NULL

6. 說明每個 table, attribute, relationship

這是一個訂房的 database。

- a. Customer 是紀錄顧客資料，有顧客編號 c_id、顧客名稱 c_name、顧客電話 phone。
- b. 顧客會訂房 reservation 送出訂單 orders，orders 也會給顧客收據 receipt，同一名顧客可以送出多筆訂房與收到多筆收據。
- c. Orders 為訂房訂單，裡頭有訂單編號 o_id、入住日期 chek_in_day、退房日期 check_out_day、入住人數 num_of_people。
- d. 顧客會參加 participation 飯店提供的活動 activity 如游泳，或是飯店幫忙顧客預定的活動如浮潛、潛水等等。其中有參加人數 num_of_participant。可以是多個顧客參與多種活動。
- e. activity 是顧客參與的活動，有活動名稱 act_name、活動價格 act_price、時間 time。
- f. 由飯店員工安排 arrange 活動，可由一個員工辦理多項活動的報名。
- g. Employee 是飯店員工。有員工編號 e_id、員工姓名 e_name、員工部門 department。
- h. 多筆訂單會由一位員工處理 service，並能為多筆訂單提供多個房間。會記錄房間是否完成整理 finish_or_not 可以入住。
- i. Room 是飯店房間，有房間編號 r_id、房間種類 type、房間價格 price。