

# SQLite-03.建立資料庫的規則

## 大數據分析

- R/Python/Julia/SQL程式設計與應用  
(R/Python/Julia/SQL Programming and Application)
- 資料視覺化 (Data Visualization)
- 機器學習 (Machine Learning)
- 統計品管 (Statistical Quality Control)
- 最佳化 (Optimization)



**李明昌** 博士

[alan9956@gmail.com](mailto:alan9956@gmail.com)

<http://rwepa.blogspot.com/>

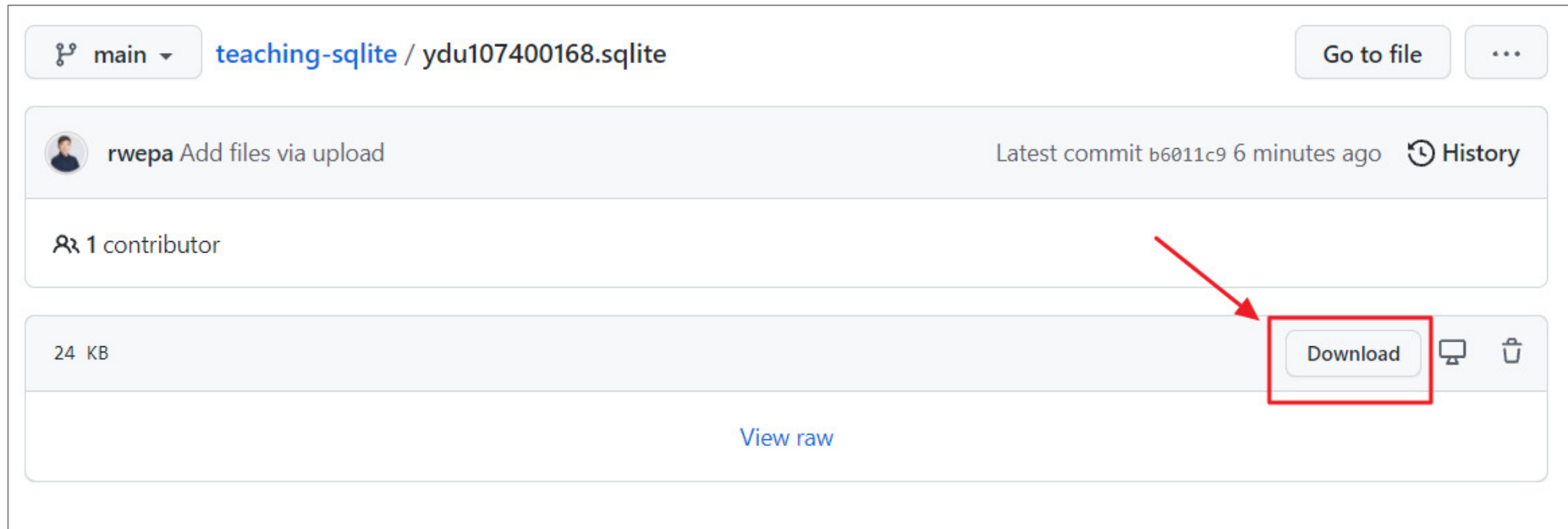
# 大綱

- 3-1 關聯式資料庫
- 3-2 鍵值屬性
- 3-3 關聯式資料庫的類型
- 3-4 關聯式資料庫的規則

參考資料：SQLite關聯式資料庫-使用SQLiteStudio結合App Inventor 2實作, 李春雄, 出版社：台科大

# 下載 ydu107400168.sqlite

- <https://github.com/rwepa/teaching-sqlite/blob/main/ydu107400168.sqlite>



The screenshot shows the GitHub interface for the file `ydu107400168.sqlite` in the `teaching-sqlite` repository. The file size is 24 KB. A red arrow points to the `Download` button, which is highlighted with a red box. Other visible elements include the repository name, the latest commit information, and the contributor list.

main ▾ teaching-sqlite / ydu107400168.sqlite Go to file ...

rwepa Add files via upload Latest commit b6011c9 6 minutes ago History

1 contributor

24 KB Download View raw

# 3-1 關聯式資料庫

---

# 關聯式資料庫

- 關聯式資料庫(Relational database)，是建立在關聯模型基礎上的資料庫，藉由集合代數等數學概念和方法來處理資料庫中的資料。
- 現實世界中的實體與實體之間的各種關係可以使用關聯模型來表示，關聯模型是由埃德加·科德(E. F. Codd)於1970年首先提出。
- 二個資料表之間可以使用相同的欄位來產生關連。
  - 例:學生資料表(學號, 姓名)  $\leftrightarrow$  選課資料表(課程編號, 開課班級, 學號)
  - 學生資料表(學號)-表示主鍵 (Primary Key, PK)
  - 選課資料表(學號)-表示外來鍵 (Foreign Key, FK)

[https://en.wikipedia.org/wiki/Relational\\_database](https://en.wikipedia.org/wiki/Relational_database)

# 關聯式資料庫的特性

- 依照資料性質的不同，資料會存放於不同資料表(Tables)。
- 資料表包括一個以上的欄位。
- 節省重複輸入的時間，提高行政效率。
- 不用重複輸入，節省儲存空間。
- 確保資料新增，修改，刪除的一致性與完整性。
- 管理關聯式資料庫的系統稱為關聯式資料庫管理系統 (Relational Database Management System, RDBMS)。

# 重複輸入的缺點

- 系名須重複輸入
- 資料可能輸入錯誤

|   | A     | B  | C    | D   |
|---|-------|----|------|-----|
| 1 | 學號    | 姓名 | 系碼   | 系名  |
| 2 | S0001 | 山松 | D001 | 資工系 |
| 3 | S0002 | 仙仙 | D001 | 資訊系 |
| 4 | S0003 | 賢蝴 | D002 | 資管系 |

# 資料表的分割

學生資料表

| 學號    | 姓名 | 系碼   |
|-------|----|------|
| S0001 | 山松 | D001 |
| S0002 | 仙仙 | D001 |
| S0003 | 賢瑚 | D002 |

科系代碼表

| 系碼   | 系名  | 系主任 |
|------|-----|-----|
| D001 | 資工系 | 李大明 |
| D002 | 資管系 | 張小君 |

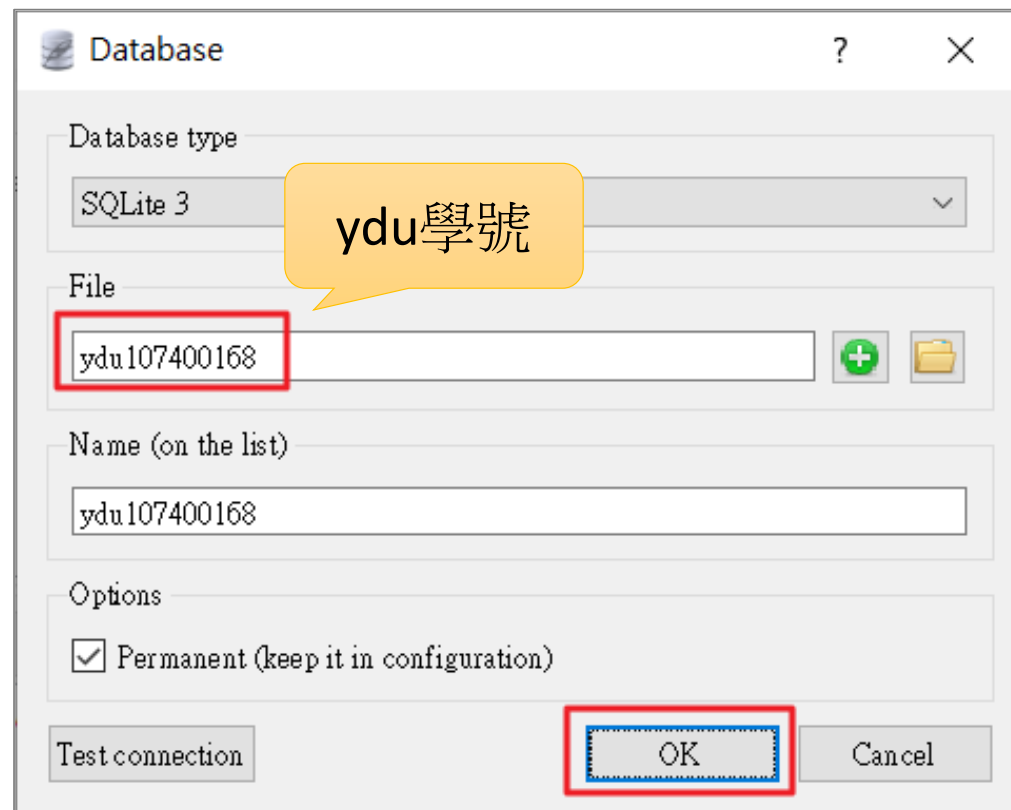
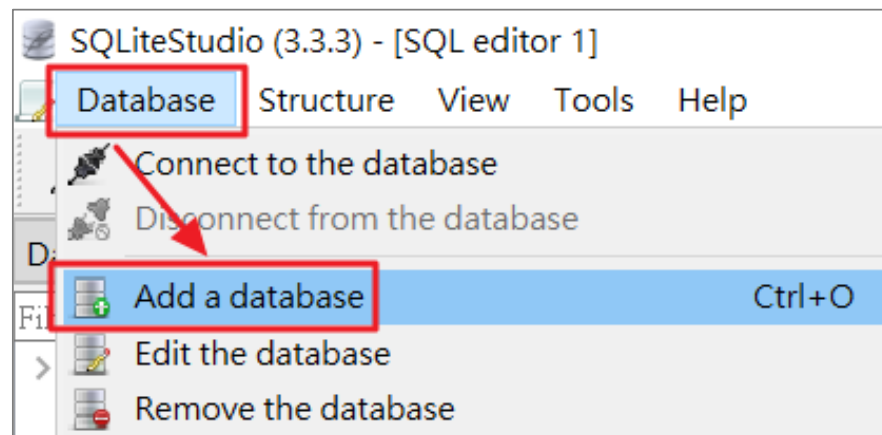
主鍵

外來鍵

- 主鍵:不可重複, 不空以空值 (NULL)
- 外來鍵:必須與另一個資料表的主鍵相同.



# 新增資料庫- ydu學號

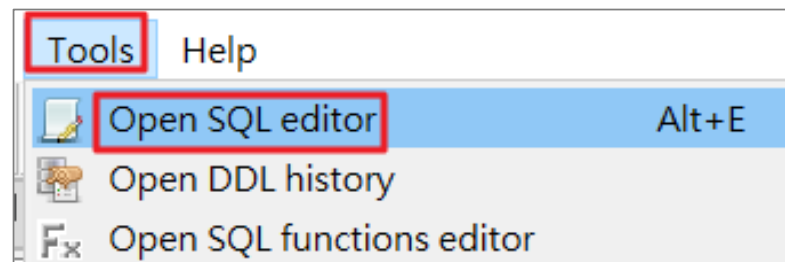


# 資料庫連線

- 選取 ydu107400168 (SQLite 3) \ 右鍵 \ Connect to the database

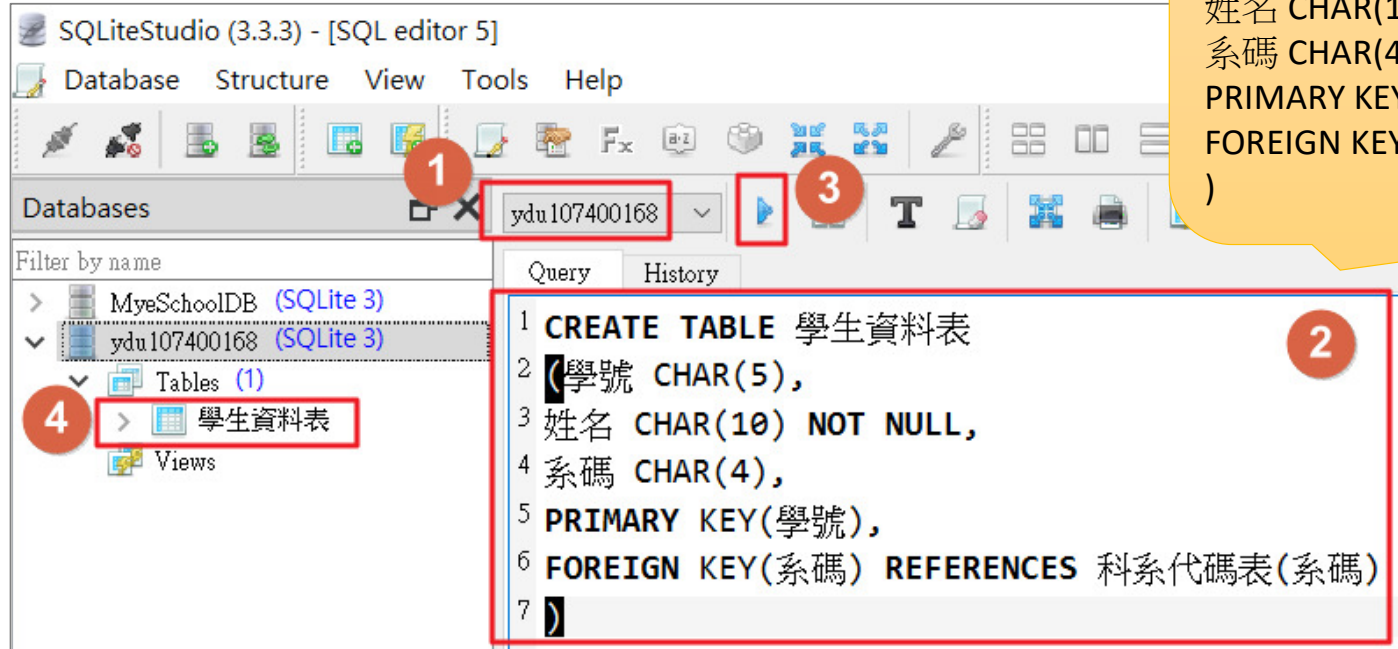


- Tools \ Open SQL editor



# CREATE TABLE 學生資料表

## • 資料定義語言 (DDL)

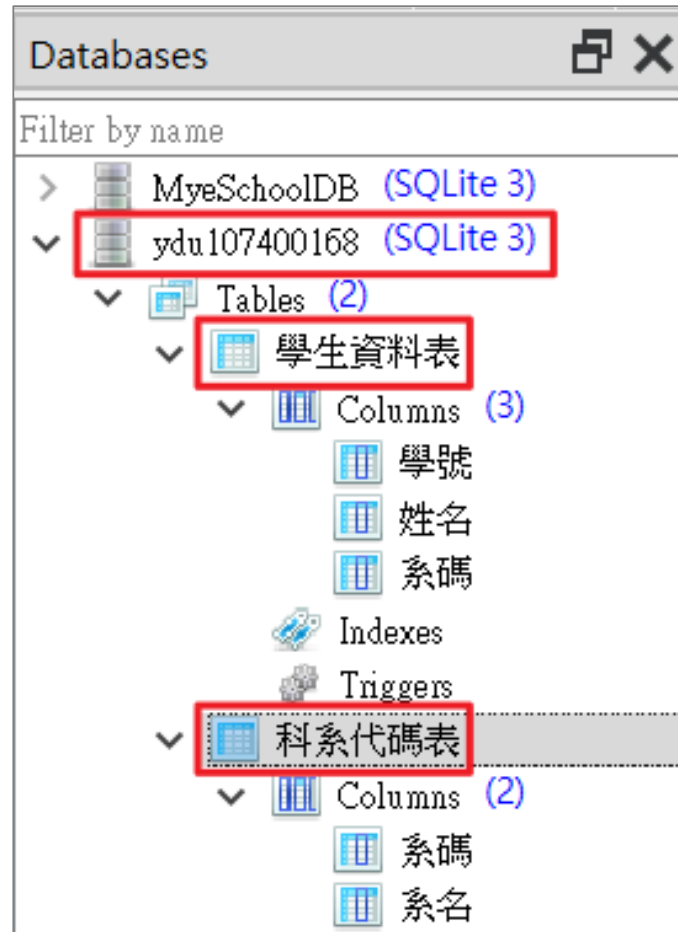


CREATE TABLE 學生資料表  
 (學號 CHAR(5),  
 姓名 CHAR(10) NOT NULL,  
 系碼 CHAR(4),  
 PRIMARY KEY(學號),  
 FOREIGN KEY(系碼) REFERENCES 科系代碼表(系碼)  
 )

# CREATE TABLE 科系代碼表

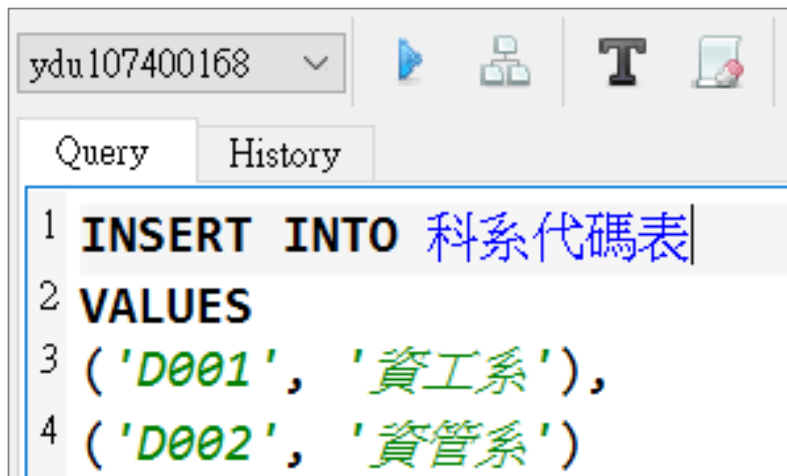
```
CREATE TABLE 科系代碼表  
(系碼 CHAR(4),  
系名 CHAR(10) NOT NULL,  
PRIMARY KEY(系碼)  
)
```

# 資料庫 ydu107400168 完成畫面

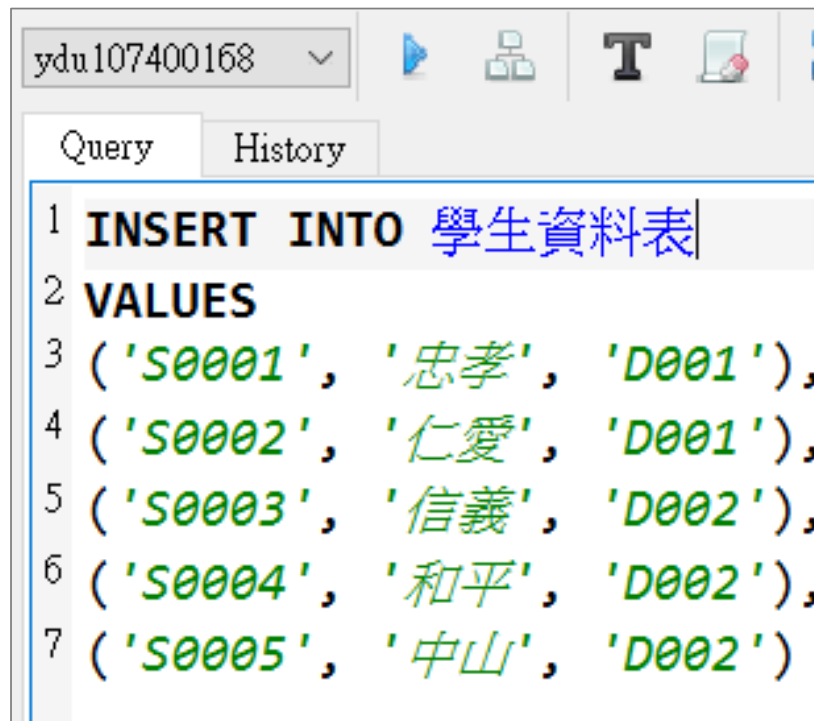


# 新增記錄 INSERT INTO

- 資料操作語言 (DML)



```
1 INSERT INTO 科系代碼表|
2 VALUES
3 ('D001', '資工系'),
4 ('D002', '資管系')
```



```
1 INSERT INTO 學生資料表|
2 VALUES
3 ('S0001', '忠孝', 'D001'),
4 ('S0002', '仁愛', 'D001'),
5 ('S0003', '信義', 'D002'),
6 ('S0004', '和平', 'D002'),
7 ('S0005', '中山', 'D002')
```

# 新增記錄 INSERT INTO (續)

INSERT INTO 科系代碼表

VALUES

('D001', '資工系'),

('D002', '資管系')

INSERT INTO 學生資料表

VALUES

('S0001', '忠孝', 'D001'),

('S0002', '仁愛', 'D001'),

('S0003', '信義', 'D002'),

('S0004', '和平', 'D002'),

('S0005', '中山', 'D002')

# 資料查詢 - SELECT

Query History

1 **SELECT \* FROM 學生資料表**

Grid view Form view

|   | 學號    | 姓名 | 系碼   |
|---|-------|----|------|
| 1 | S0001 | 忠孝 | D001 |
| 2 | S0002 | 仁愛 | D001 |
| 3 | S0003 | 信義 | D002 |
| 4 | S0004 | 和平 | D002 |
| 5 | S0005 | 中山 | D002 |

Query History

1 **SELECT \* FROM 科系代碼表**

Grid view Form view

|   | 系碼   | 系名  |
|---|------|-----|
| 1 | D001 | 資工系 |
| 2 | D002 | 資管系 |



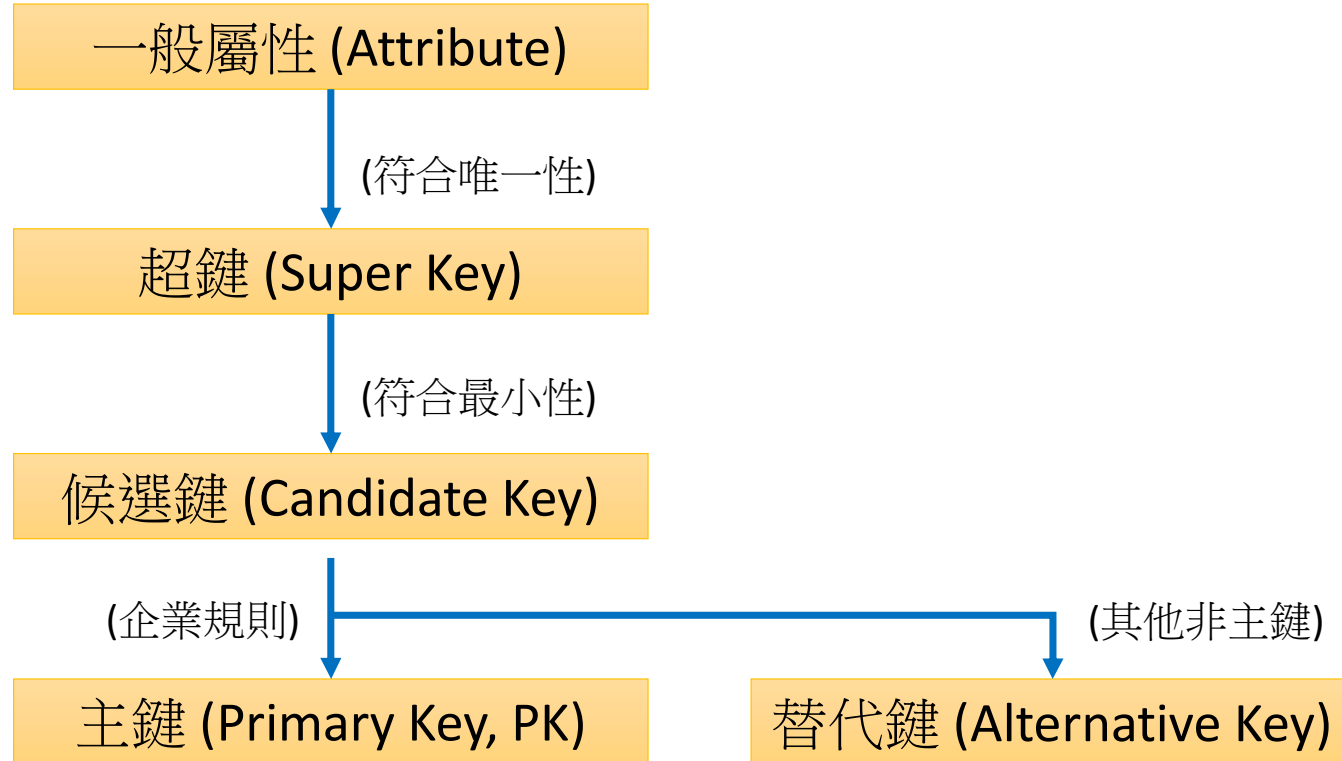
## 3-2 鍵值屬性

---

# 鍵值屬性

- 鍵值屬性(Key attribute)包括以下二大項目:
  1. 屬性(Attribute) 表示一般屬性或欄位
  2. 鍵值屬性(Key attribute) 表示由1個或1個以上的屬性所組成, 並且在一個資料表之中, 必須具有**唯一性**的屬性來當作是「鍵(key)」

# 屬性(欄位)架構



# 屬性架構-範例

範例: 學生資料表(身分證字號, 學號, 姓名, 電話, EMAIL)

- 超鍵 (Super Key)
  - (身分證字號, 學號, 姓名, 電話, EMAIL)
  - (身分證字號, 學號, 姓名, 電話)
  - (身分證字號, 學號, 姓名)
  - (身分證字號, 學號)
  - (身分證字號, 學號, 姓名, EMAIL), ...
- 候選鍵 (Candidate Key)
  - (身分證字號)
  - (學號)
- 主鍵 (Primary Key, PK)
  - (學號)
- 替代鍵 (Alternative Key)
  - (身分證字號)

# 屬性 (Attribute)

- 屬性是用來描述實體的性質. 有些屬性的值必須滿足定義範圍(定義域 domain), 例: 成績是在[0, 100]之範圍. 屬性可區分成三大類別

## 1. 簡單屬性 Simple attribute

- 已經無法再繼續分割為其他屬性
- 例: 學號

## 2. 複合屬性 Composite attribute

- 由2個或2個以上的屬性所構成
- 例: 地址: 苗栗縣 + 造橋鄉 + 談文村 + 學府路 + 168號 {縣, 鄉, 村, 路}

## 3. 衍生屬性 Derived attribute

- 經由屬性計算而新增的屬性
- 例: 銷售額 = 數量 × 單價

# 超鍵 (Super Key)

- 超鍵是在資料表中,選出2個或2個以上的欄位組合,其目的可以作為唯一識別資料用.
- 最大範圍的超鍵 → {所有欄位組合}
- 最小範圍的超鍵 → {主鍵}
- 例: {姓名, 學號} 是超鍵
- 例: {姓名, 年齡} 不是超鍵, 因為可能會有2筆資料重複

# 主鍵 (Primary Key, PK)

- 資料表中具有唯一性的欄位
- 一般會加上底線表示
  - 例: 學生資料表(學號, 姓名, 電話, EMAIL)
- 主鍵不可以重複, 例: 學號不可以重複
- 主鍵不可以為空值(NULL), 例: 學號不可以為空白
- 主鍵建議以「單一欄位」為主

# 複合鍵 (Composite Key)

- 資料表由2個或2個以上欄位所組成的唯一性欄位稱為複合鍵

複合鍵

|   | A     | B    | C  |
|---|-------|------|----|
| 1 | 學號    | 課號   | 成績 |
| 2 | S0001 | C001 | 90 |
| 3 | S0001 | C002 | 80 |
| 4 | S0002 | C001 | 60 |
| 5 | S0002 | C002 | 80 |
| 6 | S0002 | C003 | 77 |
| 7 | S0003 | C001 | 82 |

成績 會重複



# 候選鍵 (Candidate Key)

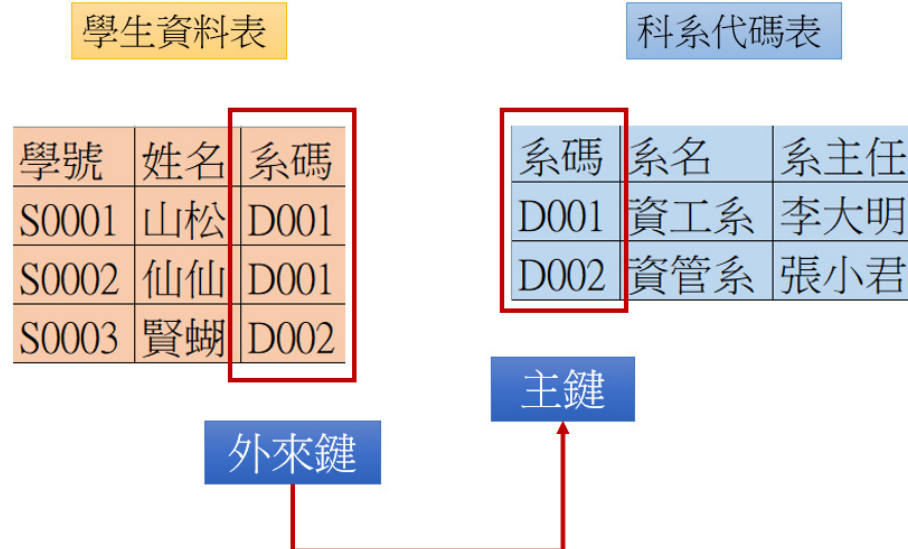
- 當有許多「主鍵」可以選擇時, 沒有被選到的主鍵稱為候選鍵
- 候選鍵的特性
  - 具有唯一性, 即不可以重複
  - 具有最小性, 即欄位個數要最少
- 例: {學號, 身分證} 具有唯一性, 但不符合最小性.
  - 因此移除{學號}之後, {身分證}是候選鍵.
  - 同理{學號}也是候選鍵.

# 外來鍵 (Foreign Key)

- 關聯式資料庫, 二個資料表進行關聯時, 必須使用外來鍵參考主鍵
- 外來鍵的資料表稱為「子關聯表」
- 主鍵的資料表稱為「父關聯表」
- 外來鍵可以找到父關聯表的主鍵記錄, 稱為資料的參考完整性

# 外來鍵的性質

- SQL使用: **學生資料表.系碼 = 科系代碼表.系碼**
- 如果「科系代碼」沒有D003, 在「學生資料表」新增D003資料時會有錯誤.
- 二個欄位的資料型態和欄位長度須相同
- 名稱可以不相同
- 外來鍵可以重複



## 3-3 關聯式資料庫的類型

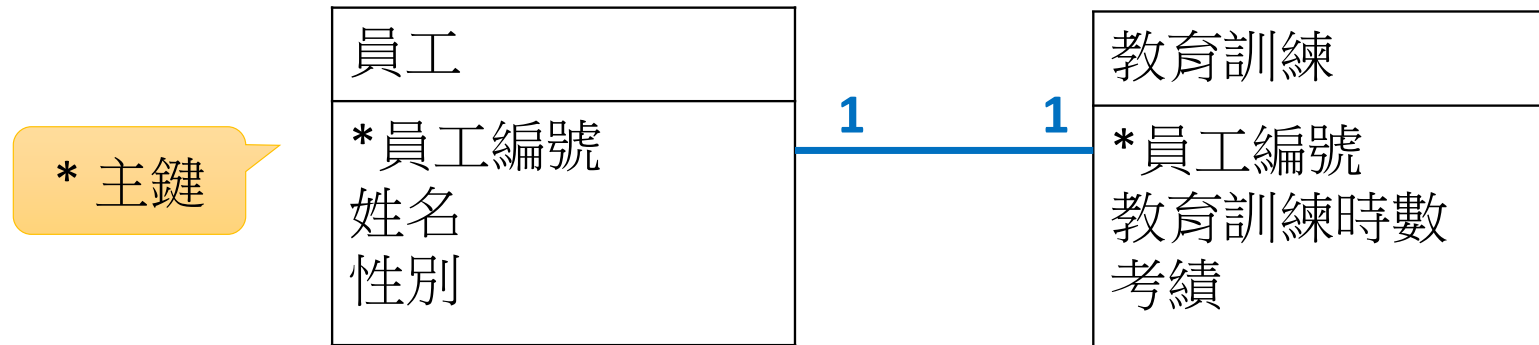
---

# 關聯式資料庫的類型

- 一對一的關聯 (1:1)
- 一對多的關聯 (1:M) – M: Many
- 多對多的關聯 (M:N)

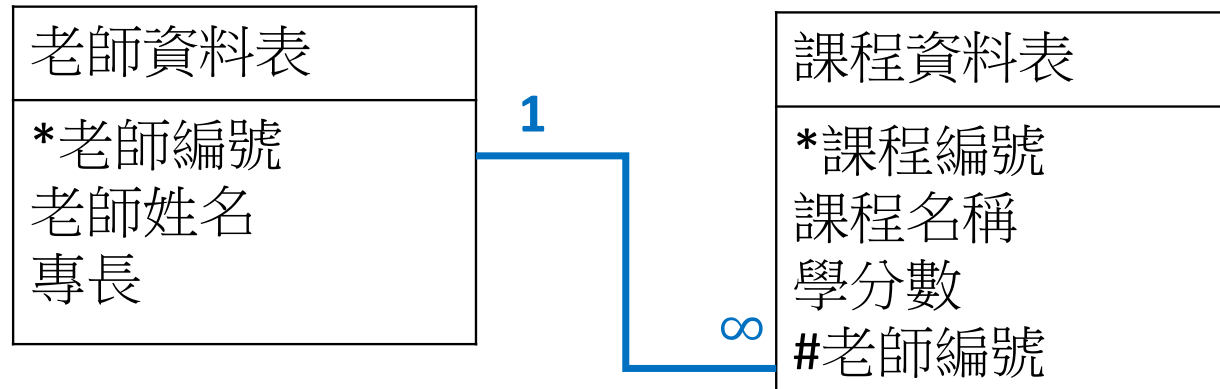
# 一對一的關聯 (1:1)

- 考慮A, B二個資料表, A資料表中的一筆記錄, 只能對應B資料表中的一筆記錄.
- 同理, B資料表中的一筆記錄, 也只能對應A資料表中的一筆記錄.
- 使用時機, 基於安全與保密考量, 將部分欄位分割到其他資料表.
- 例: 員工(員工編號, 姓名, 性別) → 教育訓練(員工編號, 教育訓練時數, 考績)
- A,B二個資料表具有一對一的關聯, 亦可合併為一個資料表.



# 一對多的關聯 (1:M)

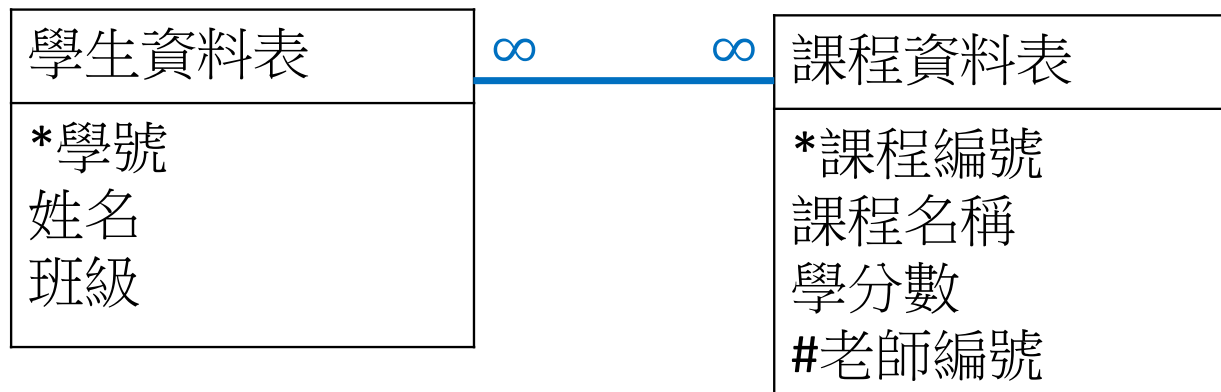
- 考慮A, B二個資料表, A資料表中的一筆記錄,可以對應B資料表中的多筆記錄.
- 但是, B資料表中的一筆記錄,只可以對應A資料表中的一筆記錄
- 例: 老師資料表 → 課程資料表
- 1位老師→可以開多門課



# 外來鍵

# 多對多的關聯 (M:N)

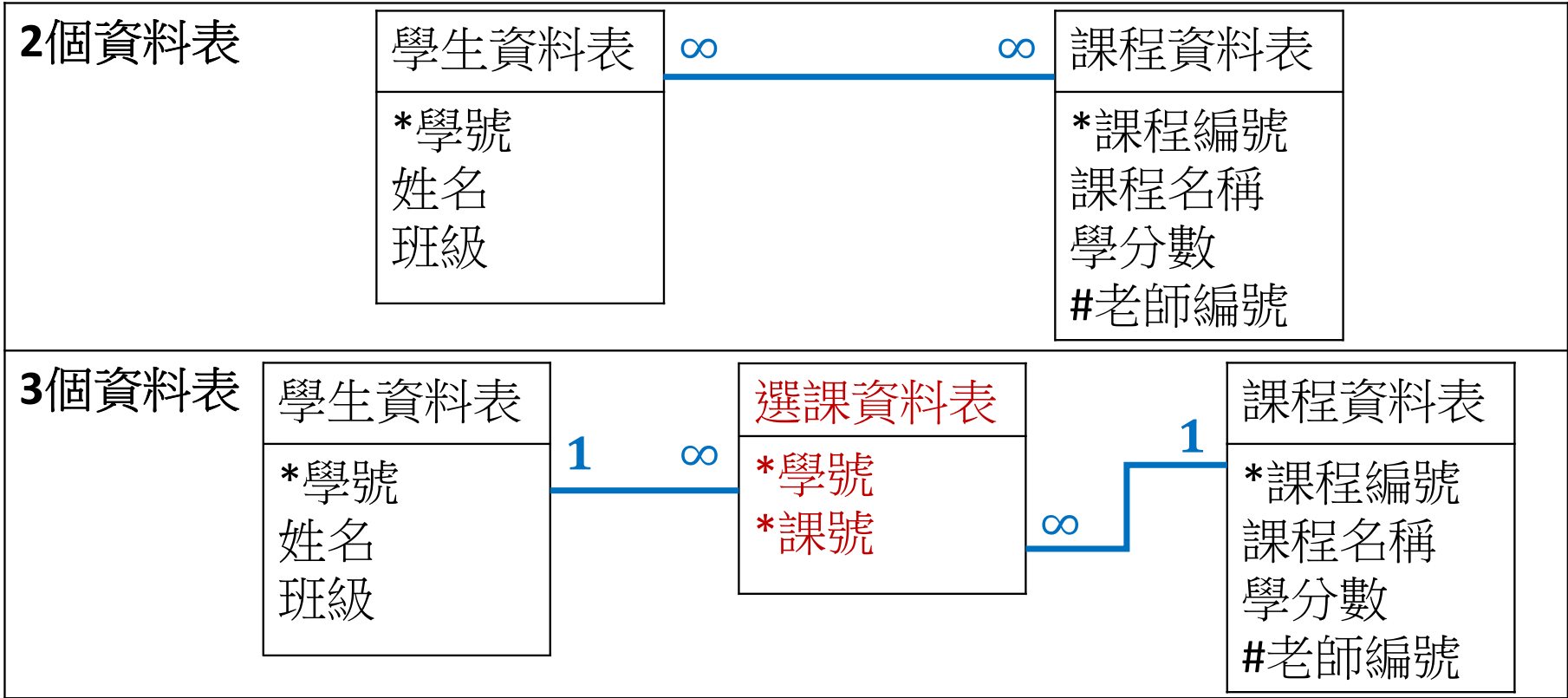
- 考慮A, B二個資料表, A資料表中的一筆記錄,可以對應B資料表中的多筆記錄.
- 同理, B資料表中的一筆記錄, 也可以對應A資料表中的多筆記錄.
- 例: 1位學生 → 選多門課; 1門課程 → 包括多位學生





# 多對多的關聯 – 分割

- 實務關聯式資料庫中, 會將2個多對多的資料表分割成3個資料表, 具有2個1對多的關聯



## 3-4 關聯式資料庫的規則

---

# 完整性規則

- 實體完整性規則 (Entity Integrity Rule)
  - 針對單一資料表, 主鍵必須具有唯一性, 且不可為空值 (NOT NULL)
- 參考完整性規則 (Referential Integrity Rule)
  - 子關聯資料表的外來鍵 (FK), 一定要存在父關聯資料表的主鍵 (PK) 之中
- 值域完整性規則 (Domain Integrity Rule)
  - 針對單一資料表, 同一資料行的資料屬性必須相同

參考: [https://en.wikipedia.org/wiki/Data\\_integrity](https://en.wikipedia.org/wiki/Data_integrity)

# 5個限制條件




- 資料庫進行新增、修改及刪除時, 會先檢查完整性規則, 包括下列5個限制條件:
  - 主鍵 Primary Key
  - 外來鍵 Foreign Key
  - 唯一鍵 Unique Key
  - 條件約束檢查 Check
  - 不可為空值 Not Null

## 5個限制條件 (續)

- 主鍵不可以為空白 (NULL)
- 值域完整性規則: 例: 成績只可以輸入 0 ~ 100
- 空值是一個特殊符號, 用以表示目前沒有資料, 空值不是「空白值」, 空值也不是「零值(zero value)」
- 非空值 (Not NULL)表示資料一定要輸入

# 外來鍵-範例

- 學生資料表(系碼)為外來鍵 (Foreign Key)




| Structure    Data    Constraints    Indexes    Triggers    DDL  |      |           |   |   |        |       |  |         |           |      |
|---|------|-----------|---|---|--------|-------|--|---------|-----------|------|
| <div> <div>ydu107400168</div> <div>Table name: 學生資料表</div> <div><input type="checkbox"/> WITHOUT ROWID</div> </div> |      |           |   |   |        |       |  |         |           |      |
|   | Name | Data type | Primary Key   | Foreign Key   | Unique | Check | Not NULL   | Collate | Generated |      |
| 1   | 學號   | CHAR (5)  |  |   |        |       |  |         |           | NULL |
| 2   | 姓名   | CHAR (10) |   |   |        |       |  |         |           | NULL |
| 3   | 系碼   | CHAR (4)  |   |  |        |       |  |         |           | NULL |

# 非空值與空值-範例

- 非空值 (Not NULL)

- 例: 系碼




ydu107400168 Table name: 學生資料表 ☐ WITHOUT ROWID

|   | Name   | Data type | Primary Key   | Foreign Key | Unique | Check | Not NULL  | Collate | Generated |
|---|--------|-----------|---|-------------|--------|-------|---|---------|-----------|
| 1 | 學號     | CHAR      |  |             |        |       |   |         | NULL      |
| 2 | 姓名     | VARCHAR   |   |             |        |       |  |         | NULL      |
| 3 | 系碼     | CHAR      |   |             |        |       |  |         | NULL      |
| 4 | E-mail | VARCHAR   |   |             |        |       |   |         | NULL      |

- 空值 (NULL)

- 例: E-mail: 可以不用輸入

ydu107400168 Table name: 學生資料表 ☐ WITHOUT ROWID

|   | Name   | Data type | Primary Key  | Foreign Key | Unique | Check | Not NULL  | Collate | Generated |
|---|--------|-----------|--|-------------|--------|-------|---|---------|-----------|
| 1 | 學號     | CHAR      |  |             |        |       |   |         | NULL      |
| 2 | 姓名     | VARCHAR   |  |             |        |       |  |         | NULL      |
| 3 | 系碼     | CHAR      |  |             |        |       |  |         | NULL      |
| 4 | E-mail | VARCHAR   |  |             |        |       |   |         | NULL      |

# 科系代碼表 - SELECT

QueryHistory

1
**SELECT**
\*
**FROM**
科系代碼表

1234

Grid viewForm view






1



Total rows loaded: 2

|   | 系碼   | 系名  |
|---|------|-----|
| 1 | D001 | 資工系 |
| 2 | D002 | 資管系 |








# 學生資料表 - SELECT




QueryHistory

1 **SELECT \* FROM 學生資料表**  
2

Grid viewForm view

1

Total rows loaded: 5

|   | 學號    | 姓名 | 系碼   |
|---|-------|----|------|
| 1 | S0001 | 忠孝 | D001 |
| 2 | S0002 | 仁愛 | D001 |
| 3 | S0003 | 信義 | D002 |
| 4 | S0004 | 和平 | D002 |
| 5 | S0005 | 中山 | D002 |

# 謝謝您的聆聽

Q & A

李明昌

*alan9956@gmail.com*

<http://rwepa.blogspot.tw/>

