

## 第十二章 外來鍵 Cascade

### 12-1 級聯 Cascade 簡介

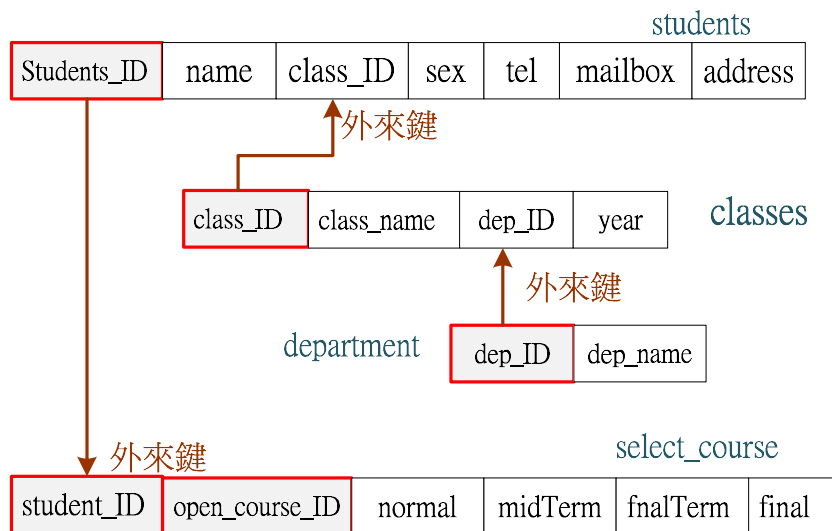
#### 12-1-1 外來鍵 Cascade 功能

『外來鍵』(Foreign Key) 表示資料表的某一欄位參考到另一資料表的某欄位，又被參考的欄位大多是『主鍵』(Primary Key)。亦是，外來鍵是跟隨另一資料表的主鍵而生，當被參考的主鍵內的『值』被刪除或更新時，外來鍵相對的該『值』的紀錄，是否隨著被刪除或更新，這就是『**Cascade**』的問題。但發生 Cascade 狀況並非僅兩個資料表之間，也許會發生在多個資料表之間。吾人以『course\_db』部分資料表之間的關聯為例，由下圖可以看出，它們之間的關聯是：

Student\_ID(select\_course) => student\_ID(students) 、

class\_ID(students) => class\_ID(classes) 、

dep\_ID(classes) => dep\_ID(department)



如表格內某欄位是屬於『外來鍵』的話，則表示其內容是隨著被參考欄位內容而改變，而改變情況有：更新、刪除動作，但表格之間可能連續幾個表格之間牽連變更，因此稱之為『**級聯**』(**Cascade**)，由上圖可能出現狀況說明如下：

#### (A) 更新動作

- 當某系所的 dep\_ID 變更(department) 時，classes 表內的 dep\_ID 是否必須隨著更新，

如沒有連帶變更的話，以後就查不出來該班是屬於哪一系所。

- 當某一班級代碼 class\_ID 變更 (classes) 時，students 表內的 class\_ID 也必須隨之更新，否則查不到該生屬於哪一班級。

### (B) 刪除動作

刪除動作可複雜了許多，範例說明如下：

- 當某位學生休退學由 students 表內刪除其 student\_ID，則相對應的 select\_course 表內有該學生的 student\_ID 也需刪除 (沒有選課了)。
- 假如某一系 (如資訊工程) 被刪除時 (由 department 的 dep\_ID)，則屬於該系的班級 (classes) 是否也需刪除，當該系的班級 class\_ID 被刪除時，則該班的學生是否也須隨之被刪除 (students 表內的 student\_ID)，跟隨著被刪除的學生選課是否也需被刪除 (select\_course 表內的 student\_ID)，此為『級聯』(Cascade) 的特質。

## 12-1-2 InnoDB 與 Cascade 語法

MySQL 資料庫的儲存引擎有兩種型態：MyISAM 與 InnoDB 兩種型態，前者是早期版本，目前較少用。對於多人使用、安全性要求較高的環境大多使用 InnoDB 型態，才具有 Cascade 之功能，目前在 MySQL 建立表格時，預設型態都指定 InnoDB 型態(功能比較請自行參考其他資料)，如下圖所示

all_course	★	瀏覽	結構	搜尋	新增	清空	刪除	16 InnoDB
classes	★	瀏覽	結構	搜尋	新增	清空	刪除	12 InnoDB
department	★	瀏覽	結構	搜尋	新增	清空	刪除	11 InnoDB
open_course	★	瀏覽	結構	搜尋	新增	清空	刪除	20 InnoDB
select_course	★	瀏覽	結構	搜尋	新增	清空	刪除	118 InnoDB
students	★	瀏覽	結構	搜尋	新增	清空	刪除	60 InnoDB
teachers	★	瀏覽	結構	搜尋	新增	清空	刪除	20 InnoDB
7 張資料表	總計							257 InnoDB

宣告 Cascade 語法如下：(在 Foreign Key 語句增加)

```
Create Table Table_name (
欄位宣告,
...,
```

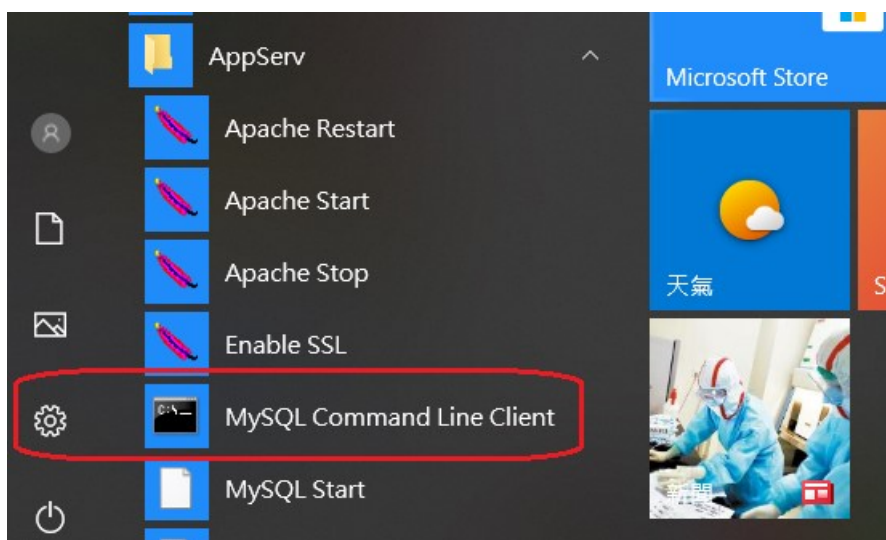
```
Foreign Key .. References .. On Delete {Cascade | Set Null | Not Action| Restrict}
Foreign Key .. References .. On Update {Cascade | Set Null | Not Action| Restrict}
) Engine InnoDB;
```

選項敘述如下：

- **On Delete** 或 **On Update**：當發生刪除或更新動作時，依後面宣告處理。
- **Cascade**：所有關聯的紀錄也會跟隨刪除或更新。
- **Set Null**：將所有關聯的紀錄設定成 Null。(對應欄位不可以 **Not Null**)
- **No Action**：即是不處理。(預設值)(系統會禁止刪除或更新動作)
- **Restrict**：與 No Action 相同。

### 12-1-3 刪除外來鍵 - AppServ

如果資料庫中表格的外來鍵沒有規劃 Cascade 功能，或設定 cascade 發生錯誤，而需刪除外來鍵重新設定時，它需要特殊動作才行 MySQL 對於刪除外來鍵有所限制，須找出所該外來鍵的 "Foreign Key id"。方法是：由桌面 => 開始 => 由選單中找出 AppServ => MySQL Command Line，如下圖所示。(如使用 mariadb 則由終端機進入 MySQL 交談介面操作)



進入交談式視窗後，輸入密碼 (12345678)，接著如下操作，則能查出 Books 與 Records 表外來鍵的識別碼。

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
mysql> use course_db;
Database changed
mysql> show create table students;
```

```

....
-----+
| students | CREATE TABLE `students` (
  `student_ID` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `name` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `class_ID` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `sex` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
  `tel` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,
  `mailbox` varchar(40) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,
  `address` varchar(60) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`student_ID`),
  KEY `class_ID` (`class_ID`),
  CONSTRAINT `students_ibfk_1` FOREIGN KEY (`class_ID`) REFERENCES `classes`
  (`class_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci |
....

```

回到 phpMyAdmin 網頁則以 SQL 命令刪除外來鍵如下：

```

Alter Table students
  Drop Foreign key `students_ibfk_1`;

```

如果 course\_db 沒有設定外來鍵，由 AppServ 功能選單內『設計』可觀察到各資料表之間都沒有聯繫，如下：(由網站下載 [course\\_db.sql](#) 的資料庫沒有建立外來鍵)

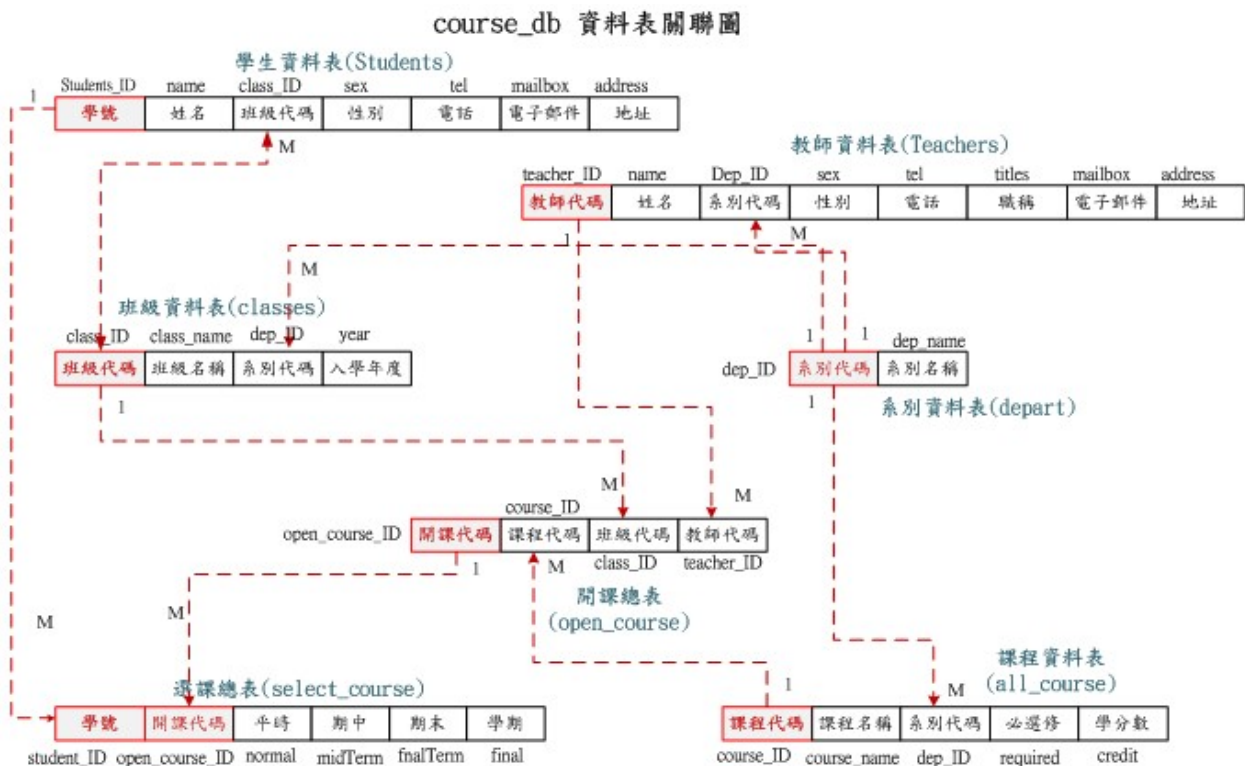
The screenshot displays several tables in the 'course\_db' database, each with its fields listed:

- students**: student\_ID (varchar(40)), name (varchar(40)), class\_ID (varchar(40)), sex (varchar(40)), tel (varchar(40)), mailbox (varchar(40)), address (varchar(60)).
- department**: Dep\_ID (varchar(40)), Dep\_name (varchar(40)).
- classes**: class\_ID (varchar(40)), class\_name (varchar(40)), Dep\_ID (varchar(40)), year (varchar(40)).
- open\_course**: open\_course\_ID (int(40)), course\_ID (varchar(40)).
- all\_course**: course\_ID (varchar(40)), course\_name (varchar(40)).
- teache**: name, Dep\_ID, sex, title, tel, mailbox, address.
- course**: student, open\_c, normal.

## 12-2 驗證 Cascade 功能 - AppServ

## 12-2-1 建立外來鍵 Cascade : course db

吾人依照『course\_db』實際需求，規劃外來鍵 Cascade 功能如下：



- 1. Not Action**：吾人設計 dep\_ID、class\_ID 不允許更新或刪除，如果允許變更的話，整個資料庫影響非常大。如真的需要刪除或更新時，則須由參考欄位變更後，再變更被參考欄位。因此，所有參考到 dep\_ID、class\_ID 欄位的外來鍵都設定為『預設值，Not Action』。
- 2. On Delete Cascade**：吾人設計 student\_ID 與 teacher\_ID 允許刪除 (cascade)，但不允許變更。譬如學生休退學或教師離職，則相關選課與開課應該隨之刪除。但變更學生學號與教師代碼沒有實質意義，而禁止更新。
- 3. On Delete Cascade On Update Cascade**：課程代碼 course\_ID 與開課代碼 open\_course\_ID 可能會隨需求而變更或刪除，我們允許它並使它有 Cascade 功能。

依照上述規範，吾人規劃外來鍵特性命令如下：(10 條)

```

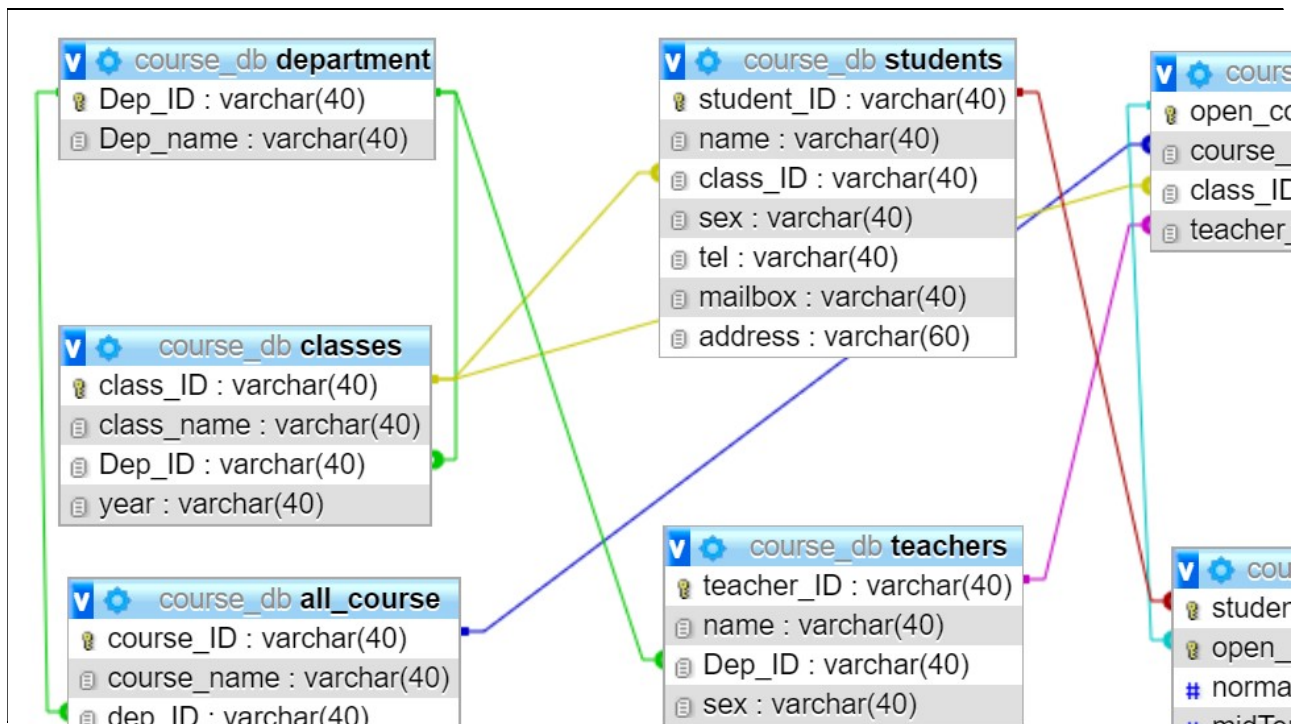
/* 第一條規範 (預設值 Not Action) 禁止 class_ID、Dep_ID 變更 */
Alter Table students ADD Foreign key (class_ID) References classes(class_ID);
Alter Table teachers ADD Foreign key (Dep_ID) References department(Dep_ID);
Alter Table classes ADD Foreign key (Dep_ID) References department(Dep_ID);
Alter Table open_course ADD Foreign key (class_ID) References classes(class_ID);
    
```

```

Alter Table open_course ADD Foreign key (class_ID) References classes(class_ID);
Alter Table all_course ADD Foreign key (Dep_ID) References department(Dep_ID);
/* 第二條規範 (On Delete Cascade) 禁止 student_ID、teacher_ID 變更但允許刪除 cascade
*/
Alter Table select_course ADD Foreign key (student_ID) References students(student_ID) On
Delete Cascade;
Alter Table open_course ADD Foreign key (teacher_ID) References teachers(teacher_ID) On Delete
Cascade;
/* 第三條規範 (On Update Cascade On Delete Cascade)允許 course_ID、open_course_ID 刪
除、更新且 cascade */
Alter Table open_course ADD Foreign key (course_ID) References all_course(course_ID) On Delete
Cascade On Update Cascade;
Alter Table select_course ADD Foreign key (open_course_ID) References
open_course(open_course_ID) On Delete Cascade On Update Cascade;

```

建立所有外來鍵之後，可觀察到各資料表之間的關係圖，如下：[\(由網站下載 course\\_db\\_foreign.sql 是已建立外來鍵的資料庫\)](#)



### 12-2-2 驗證：not Action

假設吾人欲刪除 department 表內『資訊管理系』紀錄，但它的 dep\_ID 被參考到 teachers 表的 dep\_ID 欄位，則它是被禁止刪除的 (**Not Action**)，如下：

```
// 查詢 teachers 表內有資料參考到 department 表內的 dep_ID 欄位。
select teacher_ID, name, dep_ID
from teachers
where dep_ID = (select dep_ID
                from department
                where dep_name = "資訊管理系");
```

teacher_ID	name	de
0107	張永明	11
0108	粘添壽	11

// 驗證 department 表內有關『資訊管理』(dep\_ID 欄位) 是禁止刪除的。

```
delete
from department
where dep_name = "資訊管理系";
```

**錯誤**

**SQL 查詢：**

```
delete
```

(無法刪除，表示該筆記錄有被其它表格參考到。 **Not Action** 功能)

### 12-2-3 驗證：on Delete Cascade

在 students 表的 student\_ID 被參考是允許刪除但不允許更新，吾人直接在 AppServ 上驗證其特性，如下：欄位 由 Books 表中刪除『紅樓夢』書籍，觀察 Records 是否還有人租用該本書。

- 首先查詢『林名桂』同學的選課紀錄：(僅印出學號與開課代碼)。

```
select student_ID, open_course_ID
from select_course
where student_ID = (select student_ID
                   from students
                   where name = "林名桂");
```

student_ID	open_course_ID
40011124	3
40011124	4
40011124	13

- 假設吾人欲變更其學號，觀察是否允許，如下：

```

update students
set student_ID = "4001199"
where name = "林名桂";

```

**錯誤**

SQL 查詢：

update students 表示外來鍵禁止更新

- 吾人由 students 刪除『林名桂』同學紀錄，觀察 select\_course 表是否還有它的選課紀錄，如下：

```

//查詢林名桂同學的學號
Select student_ID, name
from students
where name = "林名桂";

```

student_ID	name
40011124	林名桂

```

//刪除林名桂同學的紀錄 (students)，再觀察他的選課紀錄是否一併刪除。
delete
from students
where name = "林名桂";
select student_ID, open_course_ID
from select_course
where student_ID = "40011124";

```

student_ID	open_course_ID

傳回空值，Delete Cascade 正確



## 12-2-4 驗證：on Update Cascade

一般外來鍵允許 Update cascade 大多也允許 Delete Cascade，吾人僅驗證 Update Cascade 即可。假設吾人欲將 all\_course 表內的『電腦動畫設計』的 course\_ID 改為 90A903，觀察 open\_course 表內相關紀錄是否也跟隨著被更改。

- 首先查詢『電腦動畫設計』的 course\_ID 為何(由 all\_course 表)，再查詢該課程是否已開課 (由 open\_course 表)。

```
select *
from open_course
where course_ID = (select course_ID
                  from all_course
                  where course_name = "電腦動畫設計");
```

open_course_ID	course_ID	class_ID	te
5	90A902	902112	01

- 再由 all\_course 表將『電腦動畫設計』的 course\_ID 改為 90A903，完成後再觀察 all\_course 與 open\_course 表內該紀錄的 course\_ID 皆隨之更改 (Update Cascade)，如下：

```
// 更新 course_ID
update all_course
set course_ID = "90A903"
where course_name = "電腦動畫設計";

// 觀察 open_course 表內 course_ID 是否跟隨更新
select *
from open_course
where course_ID = (select course_ID
                  from all_course
                  where course_name = "電腦動畫設計");
```

open_course_ID	course_ID	class_ID	teacher_ID
5	90A903	902112	0114

Open\_course 表內 course\_ID 被變更，Update Cascade 正確

## 12-3 自我挑戰：Cascade

## 12-3-1 自我挑戰：刪除班級

### (A) 系統功能：PM12\_1

請在 `course_db` 資料庫上建立一只可供刪除某一班級的工具，期望如下：

- 『表單網頁』：可供下拉式選擇班級名稱，再傳送 `class_ID` 給『執行網頁』。
- 『執行網頁』：刪除該班級並顯示結果。

### (B) SQL 系統分析

如下圖所示，依照我們的規劃 `classes` 表的 `class_ID` 欄位被參考到 `students` 與 `open_course` 表內的 `classes` 欄位，並起設定為 `not Action`，表示不能直接刪除 `classes` 表內有被參考到 `class_ID` 的值，因此必須先刪除到被它參考的資料。譬如，欲刪除『日四資管三甲』（`class_ID = 400111`）班級，則需先刪除掉 `students` 表內 `class_ID = 400111` 的所有紀錄，與刪除掉 `open_course` 表內有 `class_ID = 400111` 的所有紀錄，之後才可以刪除 `classes` 表內 `class_ID = 400111`（日四資管三甲）紀錄。

