

Database Management

Homework 2

B11705044

Yen-Hung, Chiang

1

(a)

(i)

是，INSTRUCTOR 連到 DEAN 的線上標示 (0,1)，代表一個教師最多只可以當一個學院的院長。

(ii)

否，INSTRUCTOR 連到 TEACHES 的地方標示 (0,N)，代表一個教師可以不教課也可以教多個班次的課。SECTION 連到 TEACHES 的地方標示 (1,1)，代表一個班次的課必須由恰好一個教師負責。

(iii)

否，STUDENT 連到 HAS 的線上標示 (0,1)，代表一個學生最多可以隸屬一個系，也可以不隸屬任何系。DEPT 連到 HAS 的線上標示 (0,N)，代表每個學系可以擁有零位、一位或多位學生。

(iv)

是，STUDENT 連到 TAKES 這個動作的時候有標示 (0,N)，代表學生可以修零個、一個或多個班次。TAKES 連到 SECTION 的線上標示 (5,N)，代表一個班次至少要有五名學生修課。

(v)

是，因為 CoName 是 key attribute，所以不能重複。

(b)

下圖 (Figure 1) 為根據題目要求繪製出的 relational schema diagram。

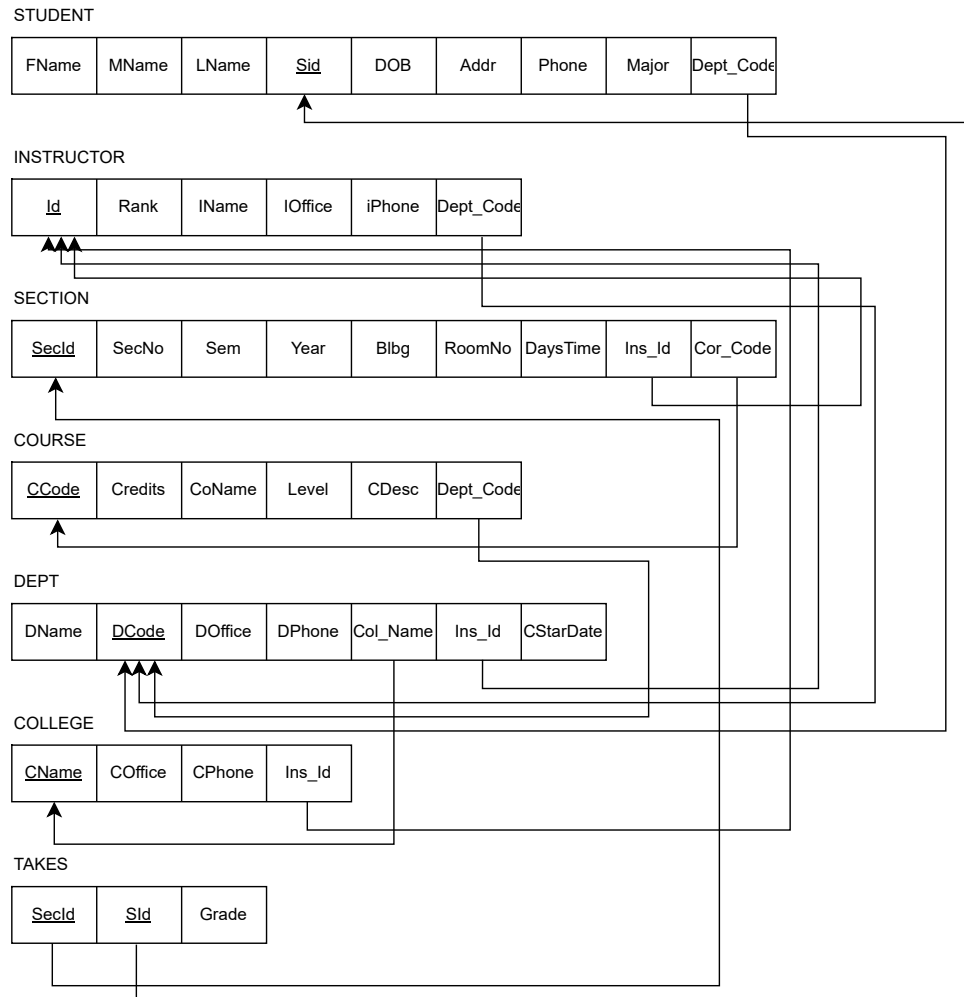


Figure 1: Relational schema diagram

(c)

這個資料庫裡應該會有七張表，其中有六張表分別對應著 STUDENT、INSTRUCTOR、DEPT、SECTION、COLLEGE、COURSE 六個 entity，另外一張表 TAKES 紀錄了 STUDENT 和 SECTION 的關係。以下是各個 entity 之間的關係。

- 每個課程可以分成好幾個不同的班次也可以不分，且每個班次都一定要屬於一門課。
- 一個學系可以提供好幾個課程也可以不提供，每門課只能屬於一個學系。
- 一個學院可以有許多個學系也可以沒有，每個學系都只屬於一個學院。
- 每個學院一定且只能有一位教師當院長，一位教師最多只能當一個學院的院長也可以不當。
- 每個學系一定且只能有一位教師當系主任，一位教師最多只能當一個學系的系主任也可以不當。
- 每個教師一定且只能被一個學系雇用，每個學系可以雇用多位教師也可以不雇用。

- 每個班次的課一定且只能被一位教師教，一位教師可以上很多班次的課也可以不上。
- 每個系可以有很多位學生也可以沒有任何學生，而每位學生最多可以屬於一個系也可以不屬於任何系。
- 最後每位學生可以選多個班次的課也可以不選，但每個班次的課至少要有五個以上的學生才成立，而 **TAKES** 就是在記錄這個關係。

2

(a)

下圖（Figure 2）為根據題目要求繪製出的實體關聯圖。圖中共有四個實體，分別為 **STUDENT**、**APPLICATION**、**LOCKER**、以及 **SEMESTER**。

- **STUDENT** 記載了學生的學號 (**Student_Id**)。
- **APPLICATION** 記載了申請編號 (**Application_Id**)、請求幾個大櫃子 (**Large_Cnt**)、幾個小櫃子 (**Small_Cnt**)、以及是否取消 (**Is_Canceled**)。
- **SEMESTER** 記載了學期編號 (**Semester_Id**)、以及每學期可用的系櫃數量 (**Available_Lockers**)。
- **LOCKER** 記載了系櫃編號 (**Locker_Id**)、系櫃尺寸 (**Size**)、以及處於何種狀態 (**Status**)。

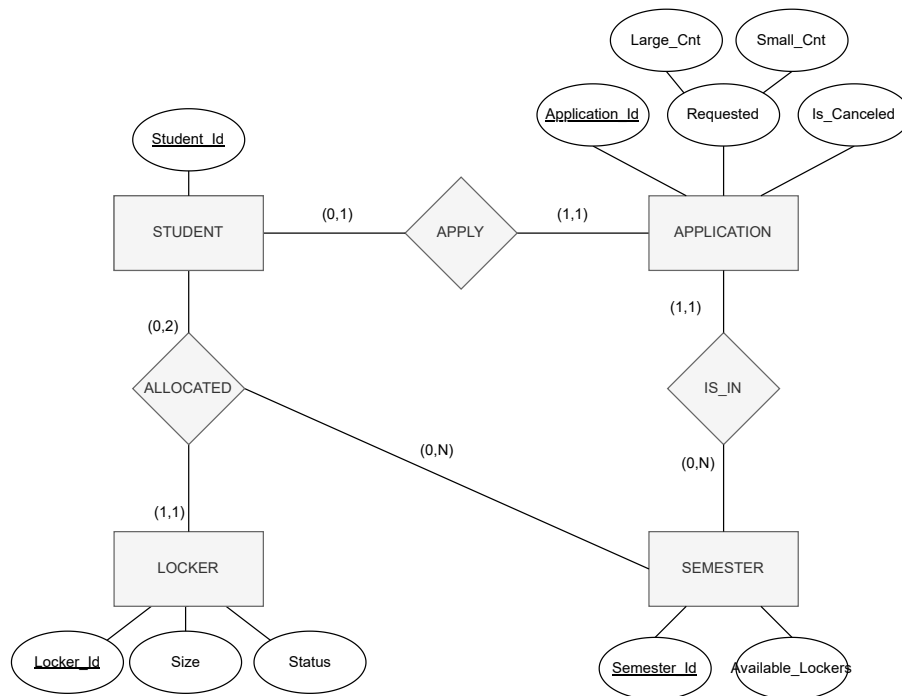


Figure 2: ER-diagram

(b)

下圖 (Figure 3) 為根據 (a) 小題繪製出來的 relational schema diagram。

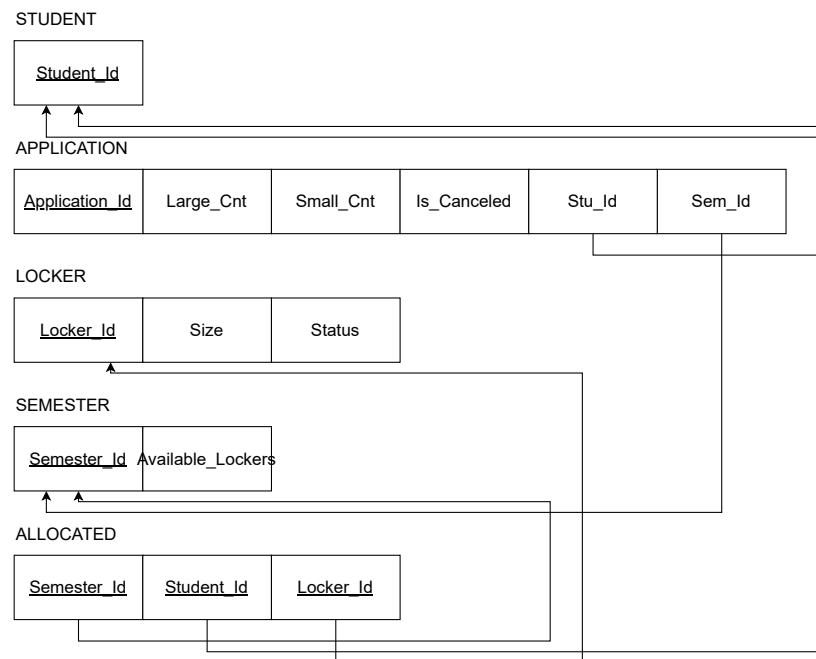


Figure 3: Relational schema diagram

(c)

下面的 SQL 程式碼 (Listing 1) 會在資料庫中產生 (b) 小題的 schema。

```
1 CREATE TABLE STUDENT(  
2     Student_Id INT NOT NULL,  
3     PRIMARY KEY(Student_Id)  
4 );  
5  
6 CREATE TABLE SEMESTER(  
7     Semester_Id INT NOT NULL,  
8     Available_Locker INT NOT NULL,  
9     PRIMARY KEY(Semester_Id)  
10 );  
11  
12 CREATE TABLE APPLICATION(  
13     Application_Id INT NOT NULL,  
14     Large_Cnt INT,  
15     Small_Cnt INT,  
16     Is_Canceled VARCHAR(10) NOT NULL,  
17     Stu_Id INT NOT NULL,  
18     Sem_Id INT NOT NULL,  
19     PRIMARY KEY(Application_Id),
```

```

20     FOREIGN KEY(Stu_Id) REFERENCES STUDENT(Student_Id),
21     FOREIGN KEY(Sem_Id) REFERENCES SEMESTER(Semeter_Id)
22 );
23
24 CREATE TABLE LOCKER(
25     Locker_Id INT NOT NULL,
26     Size_ CHAR NOT NULL,
27     Status VARCHAR(20) NOT NULL,
28     PRIMARY KEY(Locker_Id)
29 );
30
31 CREATE TABLE ALLOCATED(
32     Semeter_Id INT NOT NULL,
33     Student_Id INT NOT NULL,
34     Locker_Id INT NOT NULL,
35     PRIMARY KEY(Semeter_Id, Student_Id, Locker_Id),
36     FOREIGN KEY(Semeter_Id) REFERENCES SEMESTER(Semeter_Id),
37     FOREIGN KEY(Student_Id) REFERENCES STUDENT(Student_Id),
38     FOREIGN KEY(Locker_Id) REFERENCES LOCKER(Locker_Id)
39 );

```

Listing 1: SQL schema for the locker rental system

3

(a)

下面一句 SQL 指令（Listing 2）可以給定一個指定的學期後，列出所有被分配到兩格的學生，以及所有有申請但沒有被分配到的學生。

```

1 WITH StudentAllocations AS (
2     SELECT
3         A.Stu_Id AS Student_Id,
4         SUM(A.Large_Cnt + A.Small_Cnt) AS Total_Applications,
5         COUNT(AL.Locker_Id) AS Total_Allocated
6     FROM
7         APPLICATION A
8     LEFT JOIN
9         ALLOCATED AL ON A.Stu_Id = AL.Student_Id AND A.Sem_Id = AL.Semeter_Id
10    WHERE
11        A.Sem_Id = ? -- 指定學期
12    GROUP BY
13        A.Stu_Id
14 )
15 SELECT
16     Student_Id,
17     Total_Applications,
18     Total_Allocated

```

```

19 FROM
20     StudentAllocations
21 WHERE
22     Total_Allocated = 2
23
24 UNION ALL
25
26 SELECT
27     Student_Id,
28     Total_Applications,
29     Total_Allocated
30 FROM
31     StudentAllocations
32 WHERE
33     Total_Allocated = 0
34 ORDER BY
35     Total_Allocated DESC;

```

Listing 2: SQL for 分配原則

(b)

```

1 SELECT
2     A1.Stu_Id AS Student_Id
3 FROM
4     APPLICATION A1
5 LEFT JOIN
6     ALLOCATED AL1 ON A1.Stu_Id = AL1.Student_Id AND A1.Sem_Id = AL1.Sem_Id
7 WHERE
8     A1.Sem_Id = ? -- 指定學期
9     AND A1.Sem_Id = A1.Sem_Id - 1 -- 前一學期
10    AND AL1.Locker_Id IS NULL
11
12 UNION ALL
13
14 SELECT
15     A2.Stu_Id AS Student_Id
16 FROM
17     APPLICATION A2
18 LEFT JOIN
19     ALLOCATED AL2 ON A2.Stu_Id = AL2.Student_Id AND A2.Sem_Id = AL2.Sem_Id
20 WHERE
21     A2.Sem_Id = ? -- 指定學期
22     AND AL2.Locker_Id IS NULL
23
24 UNION ALL
25
26 SELECT

```

```

27     A3.Stu_Id AS Student_Id
28 FROM
29     APPLICATION A3
30 JOIN
31     ALLOCATED AL3 ON A3.Stu_Id = AL3.Student_Id AND A3.Sem_Id = AL3.Sem_Id
32 WHERE
33     A3.Sem_Id = ? -- 指定學期
34     AND AL3.Locker_Id IS NOT NULL
35     AND A3.Sem_Id = A3.Sem_Id - 1;

```

Listing 3: 第一句 SQL

```

1 SELECT
2     A4.Stu_Id AS Student_Id
3 FROM
4     APPLICATION A4
5 JOIN
6     ALLOCATED AL4 ON A4.Stu_Id = AL4.Student_Id AND A4.Sem_Id = AL4.Sem_Id
7 WHERE
8     A4.Sem_Id = ? -- 指定學期
9     AND AL4.Locker_Id IS NOT NULL
10    AND (SELECT COUNT(*)
11          FROM ALLOCATED AL
12          WHERE AL.Student_Id = A4.Stu_Id
13          AND AL.Semeter_Id = A4.Sem_Id - 1) = 1 -- 前一學期被分配到恰好一格
14
15 UNION ALL
16
17 SELECT
18     A5.Stu_Id AS Student_Id
19 FROM
20     APPLICATION A5
21 JOIN
22     ALLOCATED AL5 ON A5.Stu_Id = AL5.Student_Id AND A5.Sem_Id = AL5.Sem_Id
23 WHERE
24     A5.Sem_Id = ? -- 指定學期
25     AND AL5.Locker_Id IS NOT NULL
26     AND (SELECT COUNT(*)
27          FROM ALLOCATED AL
28          WHERE AL.Student_Id = A5.Stu_Id
29          AND AL.Semeter_Id = A5.Sem_Id - 1) = 2; -- 前一學期被分配到恰好兩格

```

Listing 4: 第二句 SQL

(c)

下面兩句 SQL 指令分別可以讓學生修改申請 (Listing 5)、和取消申請 (Listing 6)。

```

1 UPDATE APPLICATION

```

```
2 SET Large_Cnt = ?, -- 申請數量
3     Small_Cnt = ? -- 申請數量
4 WHERE Stu_Id = ? -- 學號
5     AND Sem_Id = ? -- 當學期的學期 Id
6     AND Is_Canceled = 'No';
```

Listing 5: 學生修改申請

```
1 UPDATE APPLICATION
2 SET Is_Canceled = 'Yes'
3 WHERE Stu_Id = ? -- 學號
4     AND Sem_Id = ? -- 當學期的學期 Id
5     AND Is_Canceled = 'No';
```

Listing 6: 學生取消申請