

數據庫系統 SSD7 實驗報告

軟件 1804 8209180438 黃柏曛

2020 年 8 月 7 日

目录

I 實驗一、數據庫與表的基本操作	2
0.1 實驗目的	3
0.2 實驗內容	3
0.2.1 创建学生-课程数据库	3
0.2.2 創建 Table	3
0.3 實驗方法與實驗步驟	4
0.4 實驗結果	4
0.4.1 在 ArchLinux 上配置好 MySQL	4
0.4.2 創建 student_data database	5
0.4.3 將 Table 創建出來	5
0.4.4 新增屬性列與刪除屬性列	8
0.4.5 SQL 的備份與還原	10
0.5 實驗小結	13
II 實驗二、數據庫查詢與更新	14
0.6 實驗目的	15
0.7 實驗內容	15
0.7.1 在表 S, C, SC 上完成以下查詢	15
0.7.2 在表 S、C、SC 中完成下列更新	16
0.8 實驗方法與實驗步驟	17
0.8.1 查詢	17
0.8.2 插入	17
0.9 實驗結果	17

0.9.1	查詢	17
0.9.2	更新	27
0.10	實驗小結	38
III	實驗三、視圖與完整性約束	39
0.11	實驗目的	40
0.12	實驗內容	40
0.12.1	以 S , C , SC 表为基础完成以下视图定义及使用	40
0.12.2	使用 SQL 进行数据完整性控制：包括三类完整性、 check 短语、constrain 子句	41
0.13	實驗方法與實驗步驟	41
0.14	實驗結果	42
0.14.1	View 的基本操作	42
0.14.2	数据完整性控制	48
0.15	實驗小結	50
IV	實驗四、存儲過程	51
0.16	實驗目的	52
0.17	實驗內容	52
0.18	實驗方法與實驗步驟	52
0.19	實驗結果	52
0.20	實驗小結	54
V	實驗五、觸發器游標	55
0.21	實驗目的	56
0.22	實驗內容	56
0.23	實驗方法與實驗步驟	56
0.24	實驗結果	56
0.25	實驗小結	58

Part I

實驗一、數據庫與表的基本操作

0.1 實驗目的

- 熟练掌握一种 DBMS 的使用方法，完成数据库的创建、删除和连接；数据表的建立、删除；表结构的修改。
- 加深对表的实体完整性、参照完整性和用户自定义完整性的理解。

0.2 實驗內容

以下实验中，使用学生-课程数据库，它描述了学生的基本信息、课程的基本信息及学生选修课程的信息。用 SQL 语句分别完成。

0.2.1 创建学生-课程数据库

数据文件名为 *student_data*、大小 10M，日志文件名为 *student_log*、大小 5M 的新数据库，该数据库名为 *student_xxxxxx*，xxxxxx 表示班级学号。1801 班 12 号，数据库名为 *student_180112*。

0.2.2 創建 Table

学号	姓名	性别	年龄	所在系
Sno	Sname	Ssex	Sage	sdept

(1) 创建学生关系表 S

课程号	课程名	先行课	学分
Cno	Cname	Cpno	ccredit

(2) 创建课程关系表 C

学号	课程号	成绩
Sno	Cno	grade

(3) 创建学生-课程表 SC

- 将以上创建表 S、C、SC 的 SQL 命令以.SQL 文件的形式保存在磁盘上。在表中加入至少 4 个元组，第一个为本人信息。
- 在表 S 上增加“出生日期”与“身高”属性列。
- 删除表 S 的“身高”属性列。
- 备份数据库，再还原。

0.3 實驗方法與實驗步驟

- 创建数据库或连接已建立的数据库。
- 在当前数据库上建立新表。
- 定义表的结构。
- 修改表的结构。

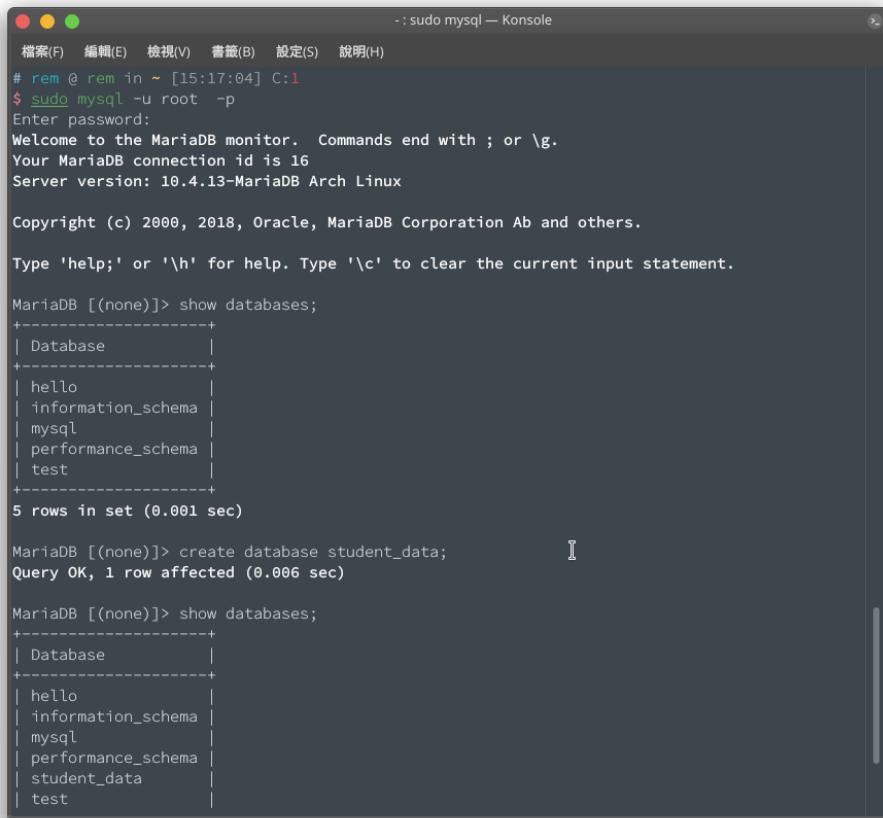
0.4 實驗結果

0.4.1 在 ArchLinux 上配置好 MySQL

```
$ sudo pacman -S mariadb # 安裝 mariadb
$ chattr +C /var/lib/mysql # disabling Copy-on-Write，因為我的文件系統是
                           Btrfs
$ systemctl enable mariadb # 啟用
$ systemctl start mariadb # 執行
$ sudo mysql -u root -p # 以 root 身份進行
```

可以自己設定一些密碼或是新增使用者

0.4.2 創建 student_data database



```
# rem @ rem in ~ [15:17:04] C:1
$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 16
Server version: 10.4.13-MariaDB Arch Linux

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| hello    |
| information_schema |
| mysql    |
| performance_schema |
| test     |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> create database student_data;
Query OK, 1 row affected (0.006 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| hello    |
| information_schema |
| mysql    |
| performance_schema |
| student_data |
| test     |
+-----+
```

(4) 創建 database

0.4.3 將 Table 創建出來

至少每個 Table 填入四筆數據

```
-: sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
-> values(1234567812, '我好棒', '人妖', 99, '軟件工程'),
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'values(1234567812, '我好
棒', '人妖', 99, '軟件工程'),' at line 3
MariaDB [student_data]> insert into S(Sno, Sname, Ssex, Sage, sdept) values(8209180438, '黃柏暉', '男', 20, '軟件工程'), (1234567812, '我好棒', '人妖', 99, '軟件工程');
Query OK, 2 rows affected (0.012 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept |
+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏暉 | 男 | 20 | 軟件工程 |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [student_data]> insert into S(Sno, Sname, Ssex, Sage, sdept) values(6666666666, '台灣NO1', '女', 18, '資訊工程'), (6666666666, '菜月昴', '男', 19, '軟件工程');
Query OK, 2 rows affected (0.007 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept |
+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏暉 | 男 | 20 | 軟件工程 |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 |
| 6666666666 | 台灣NO1 | 女 | 18 | 資訊工程 |
| 6666666666 | 菜月昴 | 男 | 19 | 軟件工程 |
+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [student_data]>
```

(5) 創建 Table S

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)

MariaDB [student_data]> describe C;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Cno   | varchar(20) | YES |   | NULL    |       |
| Cname  | varchar(10)  | YES |   | NULL    |       |
| Cpno   | varchar(10)  | YES |   | NULL    |       |
| ccredit | int(11)    | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [student_data]> insert into C (Cno, Cname, Cpno, ccredit)
-> values(bb121212, "俊俊與遙遙的日常", "BL", 10),
-> (bb131313, "Vue網頁開發", "ssd1", 4),
-> (hh202020, "flask後端", "Python", 2),
-> (jjjjjjjj, "大海掀翻小池塘", "這個宇宙太瘋狂", 8);
ERROR 1054 (42S22): Unknown column 'bb121212' in 'field list'
MariaDB [student_data]> insert into C (Cno, Cname, Cpno, ccredit) values('bb121212', "俊俊與遙遙的日常", "BL", 10), ('bb131313', "Vue網頁開發", "ssd1", 4), ('hh202020', "flask後端", "Python", 2), ('jjjjjjjj', "大海掀翻小池塘", "這個宇宙太瘋狂", 8);
Query OK, 4 rows affected (0.007 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [student_data]> select * from C;
+-----+-----+-----+-----+
| Cno   | Cname      | Cpno    | ccredit |
+-----+-----+-----+-----+
| bb121212 | 俊俊與遙遙的日常 | BL      | 10      |
| bb131313 | Vue網頁開發 | ssd1    | 4       |
| hh202020 | flask後端   | Python  | 2       |
| jjjjjjjj | 大海掀翻小池塘 | 這個宇宙太瘋狂 | 8       |
+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]>
```

(6) 創建 Table C

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from C;
+-----+-----+-----+
| Cno | Cname | Cpno | ccredit |
+-----+-----+-----+
| bb121212 | 俊俊與遙遙的日常 | BL | 10 |
| bb131313 | Vue網頁開發 | ssd1 | 4 |
| hh202020 | fLask後端 | Python | 2 |
| jjjjjjjj | 大海掀翻小池塘 | 這個宇宙太瘋狂 | 8 |
+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]> create table SC(Sno varchar(20), Cno varchar(20), grade int);
Query OK, 0 rows affected (0.026 sec)

MariaDB [student_data]> insert into SC(Sno, Cno, grade)
    -> values(8209180538, "bb121212", 100),
    -> (8209180438, "bb131313", 100),
    -> (6666666666, "jjjjjjjj", 100),
    -> (8888888888, "hh202020", 60);
Query OK, 4 rows affected (0.008 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno | Cno | grade |
+-----+-----+-----+
| 8209180538 | bb121212 | 100 |
| 8209180438 | bb131313 | 100 |
| 6666666666 | jjjjjjjj | 100 |
| 8888888888 | hh202020 | 60 |
+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]>
```

(7) 創建 Table SC

0.4.4 新增屬性列與刪除屬性列

在 S Table 新增 SBirthday (生日) 與 SHeight (身高)。

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
| 8209180438 | bb131313 |    100 |
| 66666666666 | jjjjjjjj |    100 |
| 88888888888 | hh202020 |     60 |
+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]> alter table S add column (SBirthday int, SHeight int);
Query OK, 0 rows affected (0.125 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [student_data]> describe S;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Sno   | varchar(20) | YES |   | NULL   |       |
| Sname | varchar(10)  | YES |   | NULL   |       |
| Ssex  | varchar(5)   | YES |   | NULL   |       |
| Sage  | int(11)      | YES |   | NULL   |       |
| sdept | varchar(20)  | YES |   | NULL   |       |
| SBirthday | int(11) | YES |   | NULL   |       |
| SHeight | int(11)      | YES |   | NULL   |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+-----+-----+
| Sno   | Sname | Ssex | Sage | sdept      | SBirthday | SHeight |
+-----+-----+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛 | 男   | 20 | 軟件工程 |      NULL |      NULL |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 |      NULL |      NULL |
| 66666666666 | 台灣NO1 | 女   | 18 | 資訊工程 |      NULL |      NULL |
| 66666666666 | 菜月昂 | 男   | 19 | 軟件工程 |      NULL |      NULL |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]>
```

(8) Table S 添加列

刪除 S Table 中的 SHeight (身高)。

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
+-----+-----+-----+-----+-----+
| Sname | varchar(10) | YES | | NULL |
| Ssex  | varchar(5)  | YES | | NULL |
| Sage   | int(11)    | YES | | NULL |
| sdept  | varchar(20) | YES | | NULL |
| SBirthday | int(11) | YES | | NULL |
| SHeight | int(11)  | YES | | NULL |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno   | Sname  | Ssex  | Sage  | sdept   | SBirthday | SHeight |
+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛 | 男    | 20   | 軟件工程 | NULL      | NULL    |
| 1234567812 | 我好棒 | 人妖  | 99   | 軟件工程 | NULL      | NULL    |
| 66666666666 | 台灣NO1 | 女    | 18   | 資訊工程 | NULL      | NULL    |
| 66666666666 | 菜月昂 | 男    | 19   | 軟件工程 | NULL      | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]> alter table S drop SHeight;
Query OK, 0 rows affected (0.010 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno   | Sname  | Ssex  | Sage  | sdept   | SBirthday |
+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛 | 男    | 20   | 軟件工程 | NULL      |
| 1234567812 | 我好棒 | 人妖  | 99   | 軟件工程 | NULL      |
| 66666666666 | 台灣NO1 | 女    | 18   | 資訊工程 | NULL      |
| 66666666666 | 菜月昂 | 男    | 19   | 軟件工程 | NULL      |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]>
```

(9) Table S 刪除列

0.4.5 SQL 的備份與還原

匯出 MySQL 文件

The screenshot shows a terminal window titled "mySQL — Dolphin" running on a Mac OS X desktop. The window has two panes: a file browser on the left and a terminal session on the right.

File Browser:

- Path: ~/Study/Code
- Selected file: student_data.sql

Terminal Session (zsh — Konsole):

```
Query OK, 0 rows affected (0.010 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept |
+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曠 | 男 | 20 | 軟件工程 |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 |
| 6666666666 | 台灣NO1 | 女 | 18 | 資訊工程 |
| 6666666666 | 菜月昴 | 男 | 19 | 軟件工程 |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

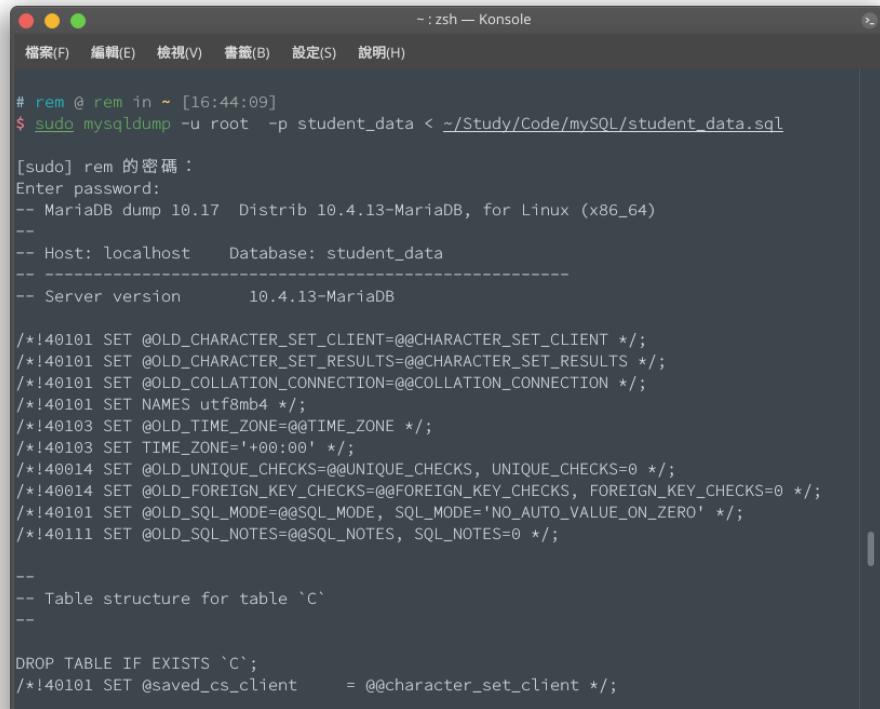
MariaDB [student_data]> exit
Bye

# rem @ rem in ~ [16:41:45]
$ sudo mysqldump -u root -p student_data > ~/Study/Code/mySQL/student_data.sql
[sudo] rem 的密碼:
Enter password:
# rem @ rem in ~ [16:44:09]
$
```

At the bottom of the terminal, there is a progress bar indicating "1 個檔案 (3.9 KiB)" and "剩餘 25.8 GiB".

(10) 匯出 student_data.sql 文件

匯入 mySQL 文件



```
# rem @ rem in ~ [16:44:09]
$ sudo mysqldump -u root -p student_data < ~/Study/Code/mySQL/student_data.sql

[sudo] rem 的密碼 :
Enter password:
-- MariaDB dump 10.17 Distrib 10.4.13-MariaDB, for Linux (x86_64)
--
-- Host: localhost      Database: student_data
-- -----
-- Server version      10.4.13-MariaDB

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `C`
--

DROP TABLE IF EXISTS `C`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
```

(11) 汇入 student_data.sql 文件

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 資源(B) 設定(S) 說明(H)
MariaDB [(none)]> use student_data;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept      | SBirthday |
+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛 | 男   | 20 | 軟件工程 | NULL |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 | NULL |
| 66666666666 | 台灣NO1 | 女   | 18 | 資訊工程 | NULL |
| 66666666666 | 菜月昂 | 男   | 19 | 軟件工程 | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [student_data]> select * from C;
+-----+-----+-----+
| Cno | Cname          | Cpno    | ccredit |
+-----+-----+-----+
| bb121212 | 俊俊與遙遙的日常 | BL     | 10 |
| bb131313 | Vue網頁開發    | ssd1   | 4  |
| hh202020 | flask後端       | Python | 2  |
| jjjjjjjjj | 大海掀翻小池塘 | 這個宇宙太瘋狂 | 8 |
+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno | Cno      | grade |
+-----+-----+-----+
| 8209180538 | bb121212 | 100 |
| 8209180438 | bb131313 | 100 |
| 66666666666 | jjjjjjjjj | 100 |
+-----+-----+-----+
```

(12) student_data 已被還原

0.5 實驗小結

本次實驗讓我們學習基本的數據庫操作，讓我們了解數據庫的基本指令，我也在這次實驗學會了怎麼使用數據庫。有時我打指令會忘記在每行指令後面添加分號，造成要重打一次，不過這樣是讓我養成習慣。

Part II

實驗二、數據庫查詢與更新

0.6 實驗目的

- 熟悉和掌握对数据表中数据的查询操作和 SQL 命令的使用，学会灵活熟练的使用 SQL 语句的各种形式，加深理解关系运算的各种操作（尤其是关系的选择，投影，连接和除运算）；
- 熟悉和掌握数据表中数据的插入、修改、删除操作和命令的使用（熟悉使用 UPDATE/INSERT/DELETE 语句进行表操作）；加深理解表的定义对数据更新的作用。

0.7 實驗內容

0.7.1 在表 S, C, SC 上完成以下查询

- 查询学生的基本信息；
- 查询“CS”系学生的基本信息；
- 查询“CS”系学生年龄不在 19 到 21 之间的学生的学号、姓名；
- 找出“CS”系年龄最大的学生，显示其学号、姓名；
- 找出各系年龄最大的学生，显示其学号、姓名；
- 统计“CS”系学生的人数；
- 统计各系学生的人数，结果按升序排列；
- 按系统计各系学生的平均年龄，结果按降序排列；
- 查询无先修课的课程的课程名和学时数；
- 统计每位学生选修课程的门数、学分及其平均成绩；
- 统计选修每门课程的学生人数及各门课程的平均成绩；
- 找出平均成绩在 85 分以上的学生成绩，结果按系分组，并按平均成绩的升序排列；

- 查询选修了“1”或“2”号课程的学生学号和姓名；
- 查询选修了课程名为“数据库系统”且成绩在60分以下的学生的学号、姓名和成绩；
- 查询每位学生选修了课程的学生信息（显示：学号，姓名，课程号，课程名，成绩）；
- 查询没有选修课程的学生的基本信息；
- 查询选修了3门以上课程的学生学号；
- 查询选修课程成绩至少有一门在80分以上的学生学号；
- 查询选修课程成绩均在80分以上的学生学号；

0.7.2 在表 S、C、SC 中完成下列更新

- 将数据分别插入表 S、C、SC；
- 将表 S、C、SC 中的数据保存在磁盘上。
- 在表 S、C、SC 上练习数据的插入、修改、删除操作。（比较在表上定义/未定义主码（Primary Key）或外码（Foreign Key）时的情况）
- 将表 S、C、SC 中的数据全部删除，再利用磁盘上备份的数据来恢复数据。
- 如果要在表 SC 中插入某个学生的选课信息（如：学号为“2007001005”，课程号为“c123”，成绩待定），应如何进行？
- 求各系学生的平均成绩，并把结果存入数据库；
- 将“CS”系全体学生的成绩置零；
- 删除“CS”系全体学生的选课记录；
- 删除学号为“S1”的相关信息；

- 将学号为“S1”的学生的学号修改为“S001”；
- 把平均成绩大于 80 分的男同学的学号和平均成绩存入另一个表 S—GRADE (SNO , AVG_GRADE)；
- 把选修了课程名为“数据结构”的学生的成绩提高 10
- 把选修了“C2”号课程，且成绩低于该门课程的平均成绩的学生成绩删除掉。

0.8 實驗方法與實驗步驟

0.8.1 查詢

- 在表 S 、 C 、 S C 上进行简单查询、连接查询、嵌套查询；
- 使用聚合函数的查询、对数据分组查询、对数据的排序查询。

0.8.2 插入

- 用 SQL 命令将数据插入当前数据库的表 S 、 C 、 S C 中；
- 用 SQL 命令形式修改表 S 、 C 、 S C 中的数据；
- 用 SQL 命令形式 删除表 S 、 C 、 S C 中的数据。

0.9 實驗結果

0.9.1 查詢

1. 查詢學生的基本訊息

```
select * from S;
```

```
MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+-----+
| Sno      | Sname    | Ssex   | Sage   | sdept    | SBirthday |
+-----+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛    | 男     | 20    | 軟件工程    | NULL      |
| 1234567812 | 我好棒    | 人妖   | 99    | 軟件工程    | NULL      |
| 66666666666 | 台灣NO1   | 女     | 18    | 資訊工程    | NULL      |
| 66666666666 | 菜月昴    | 男     | 19    | 軟件工程    | NULL      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

(13) 查询学生的基本信息

2. 查询“CS”系学生的基本信息

```
select * from S where sdept='資訊工程';
```

```
MariaDB [student_data]> select * from S where sdept='資訊工程';
+-----+-----+-----+-----+-----+-----+
| Sno      | Sname    | Ssex   | Sage   | sdept    | SBirthday |
+-----+-----+-----+-----+-----+-----+
| 66666666666 | 台灣NO1   | 女     | 18    | 資訊工程    | NULL      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.109 sec)
```

(14) 查询“CS”系学生的基本信息

3. 查询“CS”系学生年龄不在 19 到 21 之间的学生的学号、姓名。

```
select Sno, Sname from S where sdept='資訊工程' and Sage not between 19 and 21;
```

```
MariaDB [student_data]> select Sno, Sname from S where sdept='資訊工程' and Sage not between 19 and 21 ;
+-----+-----+
| Sno | Sname |
+-----+-----+
| 6666666666 | 台灣NO1 |
+-----+
1 row in set (0.262 sec)
```

(15) 查询“CS”系学生年龄不在19到21之间的学生的学号、姓名

4. 找出“CS”系年龄最大的学生，显示其学号、姓名。

這題有難度

這裡我為了實驗的操作性，將“CS”變成“軟件工程”。

```
select Sno, Sname from (select max(Sage) as Sage, Sno, Sname from S where
sdept='軟件工程');
```

```
MariaDB [student_data]> select Sno, Sname from S as A where Sage = (select max(Sage) from S where A.Sdept='
軟件工程');
+-----+-----+
| Sno | Sname |
+-----+-----+
| 1234567812 | 我好棒 |
+-----+
1 row in set (0.210 sec)
```

(16) 找出“CS”系年龄最大的学生，显示其学号、姓名

5. 找出各系年龄最大的学生，显示其学号、姓名。

```
select Sno, Sname from S as A where Sage = (select max(Sage) from S as B
where A.Sdept = B.Sdept);
```

```
MariaDB [student_data]> select Sno, Sname from S as A where Sage = (select max(Sage) from S as B where A.Sdept = B.S
dept);
+-----+-----+
| Sno | Sname |
+-----+-----+
| 1234567812 | 我好棒 |
| 6666666666 | 台灣NO1 |
+-----+
2 rows in set (0.002 sec)
```

(17) 找出各系年龄最大的学生，显示其学号、姓名

6. 统计“CS”系学生的人数。

```
select count(Sdept) from S where Sdept="軟件工程";
```

```
MariaDB [student_data]> select count(Sdept) from S where Sdept="軟件工程";
+-----+
| count(Sdept) |
+-----+
|            3 |
+-----+
1 row in set (0.218 sec)
```

(18) 统计“CS”系学生的人数

7. 统计各系学生的人数，结果按升序排列。

```
select Sdept, count(Sdept) as Sum from S group by Sdept order by Sum;
```

```
MariaDB [student_data]> select Sdept, count(Sdept) as Sum from S group by Sdept order by Sum;
+-----+-----+
| Sdept      | Sum   |
+-----+-----+
| 資訊工程    |   1  |
| 軟件工程    |   3  |
+-----+-----+
2 rows in set (0.025 sec)
```

(19) 统计各系学生的人数，结果按升序排列

8. 按系统计各系学生的平均年龄，结果按降序排列。

```
select Sdept, avg(Sage) as Age from S group by Sdept order by Age DESC;
```

```
MariaDB [student_data]> select Sdept, avg(Sage) as Age from S group by Sdept order by Age DESC;
+-----+-----+
| Sdept | Age   |
+-----+-----+
| 軟件工程 | 46.0000 |
| 資訊工程 | 18.0000 |
+-----+-----+
2 rows in set (0.001 sec)
```

(20) 按系统计各系学生的平均年龄，结果按降序排列

9. 查询无先修课的课程名和学时数。

因為實驗一的 C Table 我沒添加無先修課的數據，所以在這裡添加了一筆沒有先修課程的數據。

```
insert into C(Cno, Cname, Cpno, ccredit) values('rrrrrrrr', '我沒有先修課',
null, 100);
select Cname, ccredit from C where Cpno is null;
```

```
MariaDB [student_data]> insert into C(Cno, Cname, Cpno, ccredit) values('rrrrrrrr', '我沒有先修課', null, 100);
Query OK, 1 row affected (0.145 sec)

MariaDB [student_data]> select * from C
-> ;
+-----+-----+-----+-----+
| Cno | Cname           | Cpno | ccredit |
+-----+-----+-----+-----+
| bb121212 | 俊俊與遙遙的日常 | BL   | 10    |
| bb131313 | Vue網頁開發     | ssd1 | 4     |
| hh202020 | flask後端       | Python | 2     |
| jjjjjjjjj | 大海掀翻小池塘   | 這個宇宙太瘋狂 | 8     |
| rrrrrrrr | 我沒有先修課     | NULL  | 100   |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

(21) 插入一笔数据

```
MariaDB [student_data]> select Cname, ccredit from C where Cpno is null;
+-----+-----+
| Cname | ccredit |
+-----+-----+
| 我沒有先修課 | 100 |
+-----+-----+
1 row in set (0.015 sec)
```

(22) 查询无先修课的课程的课程名和学时数

10. 统计每位学生选修课程的门数、学分及其平均成绩。

```
select Sno, count(Cno), avg(grade) from SC as A where Sno = (select Sno from SC as B where A.Sno = B.Sno) group by Sno;
```

```
MariaDB [student_data]> select Sno, count(Cno), avg(grade) from SC as A where Sno = (select Sno from SC as B where A.Sno = B.Sno) group by Sno;
+-----+-----+-----+
| Sno | count(Cno) | avg(grade) |
+-----+-----+-----+
| 6666666666 | 1 | 100.0000 |
| 8209180438 | 1 | 100.0000 |
| 8209180538 | 1 | 100.0000 |
| 8888888888 | 1 | 60.0000 |
+-----+-----+-----+
4 rows in set (0.001 sec)
```

(23) 统计每位学生选修课程的门数、学分及其平均成绩

11. 统计选修每门课程的学生人数及各门课程的平均成绩。

```
select Cno, count(Sno), avg(grade) from SC as A where Cno = (select Cno from SC as B where A.Cno = B.Cno) group by Cno;
```

```
MariaDB [student_data]> select Cno, count(Sno), avg(grade) from SC as A where Cno = (select Cno from SC as B where A.Cno = B.Cno) group by Cno;
+-----+-----+-----+
| Cno | count(Sno) | avg(grade) |
+-----+-----+-----+
| bb121212 | 1 | 100.0000 |
| bb131313 | 1 | 100.0000 |
| hh202020 | 1 | 60.0000 |
| jjjjjjjj | 1 | 100.0000 |
+-----+-----+-----+
4 rows in set (0.001 sec)
```

(24) 统计每位学生选修课程的门数、学分及其平均成绩

12. 找出平均成绩在 85 分以上的学生成绩，结果按系分组，并按平均成绩的升序排列。

```
select S.Sno, S.sdept, AVG(SC.grade) from SC, S where SC.Sno = S.Sno group by S.Sno, S.sdept having avg(SC.grade) > 85 order by AVG(SC.grade) ASC;
```

```
MariaDB [student_data]> select S.Sno, S.sdept, AVG(SC.grade) from SC, S where SC.Sno = S.Sno group by S.Sno, S.sdept having avg(SC.grade) > 85 order by AVG(SC.grade) ASC;
+-----+-----+-----+
| Sno | sdept | AVG(SC.grade) |
+-----+-----+-----+
| 6666666666 | 軟件工程 | 100.0000 |
| 6666666666 | 資訊工程 | 100.0000 |
| 8209180438 | 軟件工程 | 100.0000 |
+-----+-----+-----+
3 rows in set (0.001 sec)
```

(25) 找出平均成绩在 85 分以上的学生成绩，结果按系分组，并按平均成绩的升序排列

13. 查询选修了“1”或“2”号课程的学生学号和姓名。

我在這裡因為沒有”1”和”2”的課程，所以改成數據庫有的。

```
select SC.Sno, S.Sname from SC, S where SC.Cno = "bb121212" and SC.Sno union
select SC.Sno, S.Sname from SC, S where SC.Cno = "bb131313" and SC.Sno =
S.Sno;
```

```

MariaDB [student_data]> select SC.Sno, S.Sname from SC, S where SC.Cno
= "bb121212" and SC.Sno union select SC.Sno, S.Sname from SC, S where
SC.Cno = "bb131313" and SC.Sno = S.Sno;
+-----+-----+
| Sno | Sname |
+-----+-----+
| 8209180538 | 黃柏曠 |
| 8209180538 | 我好棒 |
| 8209180538 | 台灣NO1 |
| 8209180538 | 菜月昴 |
| 8209180438 | 黃柏曠 |
+-----+
5 rows in set (0.001 sec)

```

(26) 查询选修了“1”或“2”号课程的学生学号和姓名

14. 查询选修了课程名为“数据库系统”且成绩在 60 分以下的学生的学号、姓名和成绩。

```

select SC.Sno, Sname, grade from SC, S, C where SC.sno = S.Sno and SC.Cno = C
.Cno and Cname = 'SSD7' and grade < 60;

```

```

MariaDB [student_data]> select SC.Sno, Sname, grade from SC, S, C where
SC.sno = S.Sno and SC.Cno = C.Cno and Cname = 'SSD7' and grade < 60;
Empty set (0.005 sec)

```

(27) 查询选修了课程名为“数据库系统”且成绩在 60 分以下的学生的学号、姓名和成绩

15. 查询每位学生选修了课程的学生信息（显示：学号，姓名，课程号，课程名，成绩）。

```

select SC.Sno, Sname, C.Cno, Cname, grade from SC, S, C where SC.Sno = S.Sno
and SC.cno = C.cno;

```

```

MariaDB [student_data]> select SC.Sno, Sname, C.Cno, Cname, grade from
  SC, S, C where SC.Sno = S.Sno and SC.cno = C.cno;
+-----+-----+-----+-----+-----+
| Sno   | Sname    | Cno     | Cname      | grade |
+-----+-----+-----+-----+
| 8209180438 | 黃柏曛    | bb131313 | Vue網頁開發 | 100  |
| 66666666666 | 台灣NO1   | jjjjjjjjj | 大海掀翻小池塘 | 100  |
| 66666666666 | 菜月昂    | jjjjjjjjj | 大海掀翻小池塘 | 100  |
+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

(28) 查询每位学生选修了课程的学生信息（显示：学号，姓名，课程号，课程名，成绩）

16. 查询没有选修课程的学生的基本信息。

```

select * from S where S.Sno not in(select SC.Sno from S, SC where SC.Sno = S.Sno);

```

```

MariaDB [student_data]> select * from S where S.Sno not in(select SC.Sno from S, SC where SC.Sno = S.Sno)
-> ;
+-----+-----+-----+-----+-----+
| Sno   | Sname    | Ssex   | Sage  | sdept   | SBirthday |
+-----+-----+-----+-----+-----+
| 1234567812 | 我好棒   | 人妖   | 99    | 軟件工程 | NULL       |
+-----+-----+-----+-----+-----+
1 row in set (0.128 sec)

```

(29) 查询没有选修课程的学生的基本信息

17. 查询选修了 3 门以上课程的学生学号。

```

select Sno from SC group by Sno having count(Cno) > 3;

```

```
MariaDB [student_data]> select Sno from SC group by Sno having count(C  
no) > 3;  
Empty set (0.000 sec)
```

(30) 查询选修了 3 门以上课程的学生学号

18. 查询选修课程成绩至少有一门在 80 分以上的学生学号。

```
select distinct Sno from SC where grade > 80;
```

```
MariaDB [student_data]> select distinct Sno from SC where grade > 80;  
+-----+  
| Sno |  
+-----+  
| 8209180538 |  
| 8209180438 |  
| 6666666666 |  
+-----+  
3 rows in set (0.000 sec)
```

(31) 查询选修课程成绩至少有一门在 80 分以上的学生学号

19. 查询选修课程成绩均在 80 分以上的学生学号。

```
select SC.Sno from SC where SC.grade > 80;
```

```
MariaDB [student_data]> select SC.Sno from SC where SC.grade > 80;  
+-----+  
| Sno |  
+-----+  
| 8209180538 |  
| 8209180438 |  
| 6666666666 |  
+-----+  
3 rows in set (0.000 sec)
```

(32) 查询选修课程成绩均在 80 分以上的学生学号

0.9.2 更新

1. 将数据分别插入表 S、C、SC。

```
- : sudo mysql - Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
-> values(1234567812, '我好棒', '人妖', 99, '軟件工程'),
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MariaDB server version for the right syntax to use near 'values(1234567812, '我好
棒', '人妖', 99, '軟件工程'),' at line 3
MariaDB [student_data]> insert into S(Sno, Sname, Ssex, Sage, sdept) values(8209180438, '黃柏暉', '男', 20, '軟件工程'), (1234567812, '我好棒', '人妖', 99, '軟件工程');
Query OK, 2 rows affected (0.012 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept |
+-----+-----+-----+-----+
| 8209180438 | 黃柏暉 | 男 | 20 | 軟件工程 |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [student_data]> insert into S(Sno, Sname, Ssex, Sage, sdept) values(6666666666, '台灣NO1', '女', 18, '資訊工程'), (6666666666, '菜月昴', '男', 19, '軟件工程');
Query OK, 2 rows affected (0.007 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept |
+-----+-----+-----+-----+
| 8209180438 | 黃柏暉 | 男 | 20 | 軟件工程 |
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 |
| 6666666666 | 台灣NO1 | 女 | 18 | 資訊工程 |
| 6666666666 | 菜月昴 | 男 | 19 | 軟件工程 |
+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [student_data]>
```

(33) 創建 Table S

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)

MariaDB [student_data]> describe C;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Cno   | varchar(20) | YES |   | NULL    |       |
| Cname  | varchar(10)  | YES |   | NULL    |       |
| Cpno   | varchar(10)  | YES |   | NULL    |       |
| ccredit | int(11)    | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [student_data]> insert into C (Cno, Cname, Cpno, ccredit)
-> values(bb121212, "俊俊與遙遙的日常", "BL", 10),
-> (bb131313, "Vue網頁開發", "ssd1", 4),
-> (hh202020, "flask後端", "Python", 2),
-> (jjjjjjjj, "大海掀翻小池塘", "這個宇宙太瘋狂", 8);
ERROR 1054 (42S22): Unknown column 'bb121212' in 'field list'
MariaDB [student_data]> insert into C (Cno, Cname, Cpno, ccredit) values('bb121212', "俊俊與遙遙的日常", "BL", 10), ('bb131313', "Vue網頁開發", "ssd1", 4), ('hh202020', "flask後端", "Python", 2), ('jjjjjjjj', "大海掀翻小池塘", "這個宇宙太瘋狂", 8);
Query OK, 4 rows affected (0.007 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [student_data]> select * from C;
+-----+-----+-----+-----+
| Cno   | Cname      | Cpno    | ccredit |
+-----+-----+-----+-----+
| bb121212 | 俊俊與遙遙的日常 | BL      | 10      |
| bb131313 | Vue網頁開發 | ssd1    | 4       |
| hh202020 | flask後端   | Python  | 2       |
| jjjjjjjj | 大海掀翻小池塘 | 這個宇宙太瘋狂 | 8       |
+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]>
```

(34) 創建 Table C

```
- : sudo mysql — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from C;
+-----+-----+-----+
| Cno | Cname | Cpno | ccredit |
+-----+-----+-----+
| bb121212 | 俊俊與遙遙的日常 | BL | 10 |
| bb131313 | Vue網頁開發 | ssd1 | 4 |
| hh202020 | fLask後端 | Python | 2 |
| jjjjjjjj | 大海掀翻小池塘 | 這個宇宙太瘋狂 | 8 |
+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]> create table SC(Sno varchar(20), Cno varchar(20), grade int);
Query OK, 0 rows affected (0.026 sec)

MariaDB [student_data]> insert into SC(Sno, Cno, grade)
    -> values(8209180538, "bb121212", 100),
    -> (8209180438, "bb131313", 100),
    -> (6666666666, "jjjjjjjj", 100),
    -> (8888888888, "hh202020", 60);
Query OK, 4 rows affected (0.008 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno | Cno | grade |
+-----+-----+-----+
| 8209180538 | bb121212 | 100 |
| 8209180438 | bb131313 | 100 |
| 6666666666 | jjjjjjjj | 100 |
| 8888888888 | hh202020 | 60 |
+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]>
```

(35) 創建 Table SC

2. 将表 S、C、SC 中的数据保存在磁盘上。

The screenshot shows a terminal window titled "mySQL — Dolphin" running on a Mac OS X desktop. The window has two panes: a top pane for "zsh — Konsole" and a bottom pane for "MySQL [student_data]".

In the MySQL pane, the user runs the command `select * from S;` which returns the following data:

Sno	Sname	Ssex	Sage	sdept	SBirthday
8209180438	黃柏曠	男	20	軟件工程	NULL
1234567812	我好棒	人妖	99	軟件工程	NULL
6666666666	台灣NO1	女	18	資訊工程	NULL
6666666666	菜月昂	男	19	軟件工程	NULL

Below the table, it says "4 rows in set (0.001 sec)". The user then types `exit` and `Bye` to exit MySQL.

After exiting MySQL, the user runs the command `sudo mysqldump -u root -p student_data > ~/Study/Code/mySQL/student_data.sql` to export the database to a file named "student_data.sql". The terminal shows the password prompt and the completion of the command.

(36) 匯出 student_data.sql 文件

3. 在表 S、C、SC 上练习数据的插入、修改、删除操作。(比较在表上定义/未定义主码 (Primary Key) 或外码 (Foreign Key) 时的情况)。

```
update S set Sname="阿俊" where Sname="菜月昂";
```

```

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+-----+
| Sno   | Sname  | Ssex  | Sage  | sdept   | SBirthday |
+-----+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛 | 男    | 20   | 軟件工程 | NULL      |
| 1234567812 | 我好棒 | 人妖  | 99   | 軟件工程 | 拍一張新 NULL |
| 66666666666 | 台灣N01 | 女    | 18   | 資訊工程 | NULL      |
| 66666666666 | 菜月昴 | 男    | 19   | 軟件工程 | NULL      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [student_data]> update S set Sname="阿俊" where Sname="菜月昴"
;
Query OK, 1 row affected (0.012 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+-----+
| Sno   | Sname  | Ssex  | Sage  | sdept   | SBirthday |
+-----+-----+-----+-----+-----+-----+
| 8209180438 | 黃柏曛 | 男    | 20   | 軟件工程 | NULL      |
| 1234567812 | 我好棒 | 人妖  | 99   | 軟件工程 | NULL      |
| 66666666666 | 台灣N01 | 女    | 18   | 資訊工程 | NULL      |
| 66666666666 | 阿俊   | 男    | 19   | 軟件工程 | NULL      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

```

(37) 更新一筆數據

<code>alter table S add primary key (Sno);</code>

```

MariaDB [student_data]> describe S;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Sno   | varchar(20) | YES  |     | NULL    |          |
| Sname | varchar(10)  | YES  |     | NULL    |          |
| Ssex  | varchar(5)   | YES  |     | NULL    |          |
| Sage  | int(11)    | YES  |     | NULL    |          |
| sdept | varchar(20) | YES  |     | NULL    |          |
| SBirthday | int(11) | YES  |     | NULL    |          |
+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [student_data]> alter table S add primary key (Sno);
Query OK, 0 rows affected (0.073 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [student_data]> describe S;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Sno   | varchar(20) | NO   | PRI | NULL    |          |
| Sname | varchar(10)  | YES  |     | NULL    |          |
| Ssex  | varchar(5)   | YES  |     | NULL    |          |
| Sage  | int(11)    | YES  |     | NULL    |          |
| sdept | varchar(20) | YES  |     | NULL    |          |
| SBirthday | int(11) | YES  |     | NULL    |          |
+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

```

(38) 設主鍵

設主鍵後，該 Value 不能為 NULL。

4. 将表 S、C、SC 中的数据全部删除，再利用磁盘上备份的数据来恢复数据。

```
drop table C;
```

```
MariaDB [(none)]> use student_data;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
                                         按一張新的螢幕

Database changed
MariaDB [student_data]> show tables;
+-----+
| Tables_in_student_data |
+-----+
| C
| S
| SC
+-----+
3 rows in set (0.000 sec)

MariaDB [student_data]> drop table C;
Query OK, 0 rows affected (0.014 sec)

MariaDB [student_data]> show tables;
+-----+
| Tables_in_student_data |
+-----+
| S
| SC
+-----+
2 rows in set (0.001 sec)
```

(39) 刪除某個資料表

```
mysqldump -u root -p student_data < ~/Study/Code/mysql/student_data.sql
```

```

# rem @ rem in ~ [22:46:48]
$ sudo mysqldump -u root -p student_data < ~/Study/Code/mySQL/student_data.sql
Enter password:
-- MariaDB dump 10.17 Distrib 10.4.13-MariaDB, for Linux (x86_64)
--
-- Host: localhost      Database: student_data
-- 
-- Server version      10.4.13-MariaDB

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=

```

(40) 還原整個 Database

- 如果要在表 SC 中插入某个学生的选课信息(如：学号为“2007001005”，课程号为 “c123”，成绩待定)，应如何进行？

```
insert into SC (Sno, Cno, grade) values("2007001005", "c123", NULL);
```

```

MariaDB [student_data]>
MariaDB [student_data]> insert into SC (Sno, Cno, grade) values("2007001005",
"c123", NULL);
Query OK, 1 row affected (0.007 sec)

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno    | Cno     | grade |
+-----+-----+-----+
| 8209180538 | bb121212 |    100 |
| 8209180438 | bb131313 |    100 |
| 6666666666 | jjjjjjjj |    100 |
| 8888888888 | hh202020 |     60 |
| 2007001005 | c123     |   NULL |
+-----+-----+-----+
5 rows in set (0.001 sec)

```

(41) 插入資料

6. 求各系学生的平均成绩，并把结果存入数据库。因為我不會一次就搞定的指令，所以只好一個一個輸。

```
select avg(grade) from SC;
insert SC(Sno, Cno, grade) values("總平均", "-", 90);
```

```
MariaDB [student_data]> select avg(grade) from SC;
+-----+
| avg(grade) |
+-----+
| 90.0000 |
+-----+
1 row in set (0.005 sec)

MariaDB [student_data]> insert SC(Sno, Cno, grade) values("總平均", "-", 90);
Query OK, 1 row affected (0.007 sec)

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno    | Cno    | grade |
+-----+-----+-----+
| 8209180538 | bb121212 | 100 |
| 8209180438 | bb131313 | 100 |
| 6666666666 | jjjjjjjj | 100 |
| 8888888888 | hh202020 | 60  |
| 2007001005 | c123    | NULL  |
| 總平均     | -        | 90   |
+-----+-----+-----+
6 rows in set (0.001 sec)
```

(42) 將平均成績插入資料

7. 将“CS”系全体学生的成绩置零。

```
update SC set grade=0 where Sno in (select SC.Sno from SC, S where SC.Sno=S.Sno and S.sdept="軟件工程");
```

```

MariaDB [student_data]> update SC set grade=0 where Sno in (select SC.Sno from
SC, S where SC.Sno=S.Sno and S.sdept="軟件工程");
Query OK, 1 row affected (0.102 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno   | Cno    | grade |
+-----+-----+-----+
| 8209180538 | bb121212 | 100 |
| 8209180438 | bb131313 | 0 |
| 6666666666 | jjjjjjjj | 100 |
| 8888888888 | hh202020 | 60 |
| 2007001005 | c123    | NULL |
| 總平均      | -        | 90 |
+-----+-----+-----+
6 rows in set (0.000 sec)

```

(43) 将“CS”系全体学生的成绩置零

8. 删除“CS”系全体学生的选课记录。

```
delete from SC where "軟件工程"=(select sdept from S where S.Sno = SC.Sno);
```

```

MariaDB [student_data]> delete from SC where "軟件工程"=(select sdept from S w
here S.Sno = SC.Sno);
Query OK, 1 row affected (0.007 sec)

MariaDB [student_data]> select * from SC;
+-----+-----+-----+
| Sno   | Cno    | grade |
+-----+-----+-----+
| 8209180538 | bb121212 | 100 |
| 6666666666 | jjjjjjjj | 100 |
| 8888888888 | hh202020 | 60 |
| 2007001005 | c123    | NULL |
| 總平均      | -        | 90 |
+-----+-----+-----+
5 rows in set (0.000 sec)

```

(44) 删除“CS”系全体学生的选课记录

9. 删除学号为“S1”的相关信息。

```
delete from S where S.Sno="S1";
```

```
MariaDB [student_data]> delete from S where S.Sno="S1";
Query OK, 0 rows affected (0.006 sec)
```

(45) 删除学号为“S1”的相关信息

10. 将学号为“S1”的学生的学号修改为“S001”。

```
update S set Sno="S001" where Sno="S1";
```

```
MariaDB [student_data]> update S set Sno="S001" where Sno="S1";
Query OK, 0 rows affected (0.000 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

(46) 将学号为“S1”的学生的学号修改为“S001”

11. 把平均成绩大于 80 分的男同学的学号和平均成绩存入另一个表 S—GRADE (SNO, AVG_GRADE)。

這題... 我理解好久:(最後參考其他同學的。

```
INSERT INTO S(Sno, grade)
SELECT SC.Sno, AVG(SC.grade) FROM SC,S
WHERE SC.Sno=S.Sno AND S.Ssex='男'
GROUP BY SC.Sno
HAVING AVG(SC.grade) > 80;
```

12. 把选修了课程名为“数据结构”的学生的成绩提高 10%。

```
UPDATE SC SET Cno=Cno*1.10
WHERE Sno IN(SELECT * FROM SC,C WHERE C.Cname='数据结构' AND SC.Cno=C.Cno);
```

13. 把选修了“C2”号课程，且成绩低于该门课程的平均成绩的学生成绩删除掉。

```
delete from SC where grade < (select avg(grade) from SC where SC.Cno="C2");
```

```
MariaDB [student_data]> delete from SC where grade < (select avg(grade) from SC where SC.Cno="C2");
Query OK, 0 rows affected (0.006 sec)
```

(47) 刪除資料

0.10 實驗小結

此次的實驗加強了我的 SQL 掌握能力，有些題目我真的想了許久一錯再錯才寫出正確的答案。這次的實驗內容也相比實驗一多很多，我每題都自己寫出來並寫成 LaTeX，蠻有成就感的。

Part III

實驗三、視圖與完整性約束

0.11 實驗目的

学习灵活熟练的进行视图的操作，认识视图的作用；掌握完整性控制的方法。

- 熟悉和掌握对数据表中视图的定义操作和 SQL 命令的使用。
- 熟悉和掌握对数据表中视图的查询操作和 SQL 命令的使用。
- 熟悉和掌握对数据表中视图的更新操作和 SQL 命令的使用，并注意视图更新与基本表更新的区别与联系。
- 熟悉 SQL 语言对数据库进行完整性控制的方法。

0.12 實驗內容

0.12.1 以 S , C , SC 表为基础完成以下视图定义及使用

- 定义“SSCH”院学生基本情况视图 V_SSCH。
- 将 S , C , SC 表中学生的学号，姓名，课程号，课程名，成绩定义为视图 V_S_C_G。
- 将各院学生人数，平均年龄定义为视图 V_NUM_AVG。
- 将各位学生选修课程的门数及平均成绩定义为视图 V_AVG_S_G 并查询结果。
- 查询平均成绩为 90 分以上的学生学号、姓名和成绩。
- 通过视图 V_SSCH，新增加一个学生记录 ('S12','YAN XI',19, 'SSCH')，并查询结果。
- 通过视图 V_SSCH，删除学号为 “S12” 学生信息，并查询结果。
- 将视图 V_SSCH 中学号为 “S12”的学生改名 “中南人”。

0.12.2 使用 SQL 进行数据完整性控制：包括三类完整性、 check 短语、constrain 子句

1. 在创建下列关系表时完成如下约束：定义实体完整性；参照完整性（外码、在删除 S 中的元组时级联删除 SC 中相应元组、当更新 S 中的 Sno 时同时更新 SC 中的 Sno）；用户定义完整性：学生年龄 <30。

2. 修改 S 中的约束条件，学号在 100 – 1000 之间。

学生关系表 S：

学号	姓名	性别	年龄	所在系
Sno	Sname	Ssex	Sage	sdept

(48) 学生关系表 S

创建课程关系表 C：

课程号	课程名	先行课	学分
Cno	Cname	Cpno	ccredit

(49) 课程关系表 C

创建学生-课程表 SC：

学号	课程号	成绩
Sno	Cno	grade

(50) 课程关系表 C

3. 用实验验证当操作违反了完整性约束时，系统如何处理？

问题：外键与参照主键是否一定要相同？

0.13 實驗方法與實驗步驟

- 定义视图、对视图进行查询、修改。

- 进行完整性约束定义、修改。
- 进行约束违例验证。

0.14 實驗結果

0.14.1 View 的基本操作

1. 定义“SSCH”院学生基本情况视图 V_SSCH。

```
create view V_SSCH as select * from S where S.sdept = "SSCH";
```

```
MariaDB [student_data]> create view V_SSCH as select * from S where S.sdept = "SSCH";
Query OK, 0 rows affected (0.009 sec)

MariaDB [student_data]> select * from V_SSCH;
Empty set (0.008 sec)
```

(51) 創建 V_SSCH

```
MariaDB [student_data]> insert into S (Sno, Sname, Ssex, Sage, sdept, SBirthday)
-> values("22222222", "我", "不知道", 22, "SSCH", null),
-> ("3333333333", "好", "你猜猜", 13, "SSCH", null),
-> ("7777777777", "菜", "無", 30, "SSCH", null)
-> ,("8888888888", "阿", "女", 8, "SSCH", null);
Query OK, 4 rows affected (0.010 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [student_data]> select * from S;
+-----+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept | SBirthday |
+-----+-----+-----+-----+-----+
| 1234567812 | 我好棒 | 人妖 | 99 | 軟件工程 | NULL |
| 2222222222 | 我 | 不知道 | 22 | SSCH | NULL |
| 3333333333 | 好 | 你猜猜 | 13 | SSCH | NULL |
| 5555555555 | 阿俊 | 男 | 19 | 軟件工程 | NULL |
| 6666666666 | 台灣NO1 | 女 | 18 | 資訊工程 | NULL |
| 7777777777 | 菜 | 無 | 30 | SSCH | NULL |
| 8209180438 | 黃柏曛 | 男 | 20 | 軟件工程 | NULL |
| 8888888888 | 阿 | 女 | 8 | SSCH | NULL |
+-----+-----+-----+-----+-----+
8 rows in set (0.000 sec)
```

(52) 插入新数据

```
MariaDB [student_data]> select * from V_SSCH;
+-----+-----+-----+-----+-----+
| Sno | Sname | Ssex | Sage | sdept | SBirthday |
+-----+-----+-----+-----+-----+
| 2222222222 | 我 | 不知道 | 22 | SSCH | NULL |
| 3333333333 | 好 | 你猜猜 | 13 | SSCH | NULL |
| 7777777777 | 菜 | 無 | 30 | SSCH | NULL |
| 8888888888 | 阿 | 女 | 8 | SSCH | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

(53) 查看 View

2. 将 S, C, SC 表中学生的学号, 姓名, 课程号, 课程名, 成绩定义为视图 V_S_C_G。

```
create or replace view V_S_C_G as select S.Sno, S.Sname, C.Cno, C.Cname, SC.
grade from S, C, SC where S.Sno = SC.Sno and C.Cno = SC.Cno;
```

```
MariaDB [student_data]> create view V_S_C_G as select S.Sno, S.Sname, C.Cno, C.Cn
ame, SC.grade from S, C, SC where S.Sno = SC.Sno;
Query OK, 0 rows affected (0.017 sec)
```

(54) V_S_C_G

因為發現自己調建設錯了，所以重新更新一次。

```
MariaDB [student_data]> create or replace view V_S_C_G as select S.Sno, S.Sname,
C.Cno, C.Cname, SC.grade from S, C, SC where S.Sno = SC.Sno and C.Cno = SC.Cno;
Query OK, 0 rows affected (0.006 sec)

MariaDB [student_data]> select * from V_S_C_G;
+-----+-----+-----+-----+-----+
| Sno | Sname | Cno | Cname | grade |
+-----+-----+-----+-----+-----+
| 6666666666 | 台灣NO1 | jjjjjjjj | 大海掀翻小池塘 | 100 |
| 8888888888 | 阿 | hh202020 | flask後端 | 60 |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)
```

(55) 修改 View

3. 将各院学生人数，平均年龄定义为视图 V_NUM_AVG。

```
create view V_NUM_AVG as select S.sdept, count(S.Sno), avg(S.Sage) from S
group by S.sdept;
```

```
MariaDB [student_data]> create view V_NUM_AVG as select S.sdept, count(S.Sno), avg(S.Sage) from S group by S.sdept;
Query OK, 0 rows affected (0.010 sec)

MariaDB [student_data]> select * from V_NUM_AVG;
+-----+-----+-----+
| sdept | count(S.Sno) | avg(S.Sage) |
+-----+-----+-----+
| SSCH | 4 | 18.2500 |
| 資訊工程 | 1 | 18.0000 |
| 軟件工程 | 3 | 46.0000 |
+-----+-----+-----+
3 rows in set (0.001 sec)
```

(56) V_NUM_AVG

4. 将各位学生选修课程的门数及平均成绩定义为视图 V_AVG_S_G 并查询结果。

```
create view V_AVG_S_G
as
select count(SC.Sno), avg(SC.grade) from SC, C
where SC.Cno = C.Cno
group by SC.Sno;
```

```

MariaDB [student_data]> create view V_AVG_S_G
-> as
-> select count(SC.Sno), avg(SC.grade) from SC, C
-> where SC.Cno = C.Cno
-> group by SC.Sno;
Query OK, 0 rows affected (0.009 sec)

MariaDB [student_data]> select * from V_AVG_S_G;
+-----+-----+
| count(SC.Sno) | avg(SC.grade) |
+-----+-----+
|           1 |      100.0000 |
|           1 |      100.0000 |
|           1 |       60.0000 |
+-----+-----+
3 rows in set (0.001 sec)

```

(57) V_AVG_S_G

5. 查询平均成绩为 90 分以上的学生学号、姓名和成绩。

```

select SC.Sno, S.Sname, SC.grade from SC, S where S.Sno = SC.Sno in (select
SC.Sno from SC group by SC.Sno having avg(SC.grade) > 90);

```

```

MariaDB [student_data]> select SC.Sno, S.Sname, SC.grade from SC, S where S.Sno =
SC.Sno in (select SC.Sno from SC group by SC.Sno having avg(SC.grade) > 90);
Empty set (0.006 sec)

```

(58) 檢詢

6. 通过视图 V_SSCH，新增加一个学生记录 ('S12','YAN XI',19,'SSCH')，并查询结果。

```

insert into V_SSCH(Sno, Sname, Sage, Sdept) values("S12", "YAN XI", 19, "SSCH");
select * from V_SSCH where Sno="S12";

```

```
MariaDB [student_data]> insert into V_SSCH(Sno, Sname, Sage, Sdept) values("S12",
  "YAN XI", 19, "SSCH");
Query OK, 1 row affected (0.008 sec)

MariaDB [student_data]> select * from V_SSCH where Sno="S12";
+----+----+----+----+----+
| Sno | Sname | Ssex | Sage | sdept | SBirthday |
+----+----+----+----+----+
| S12 | YAN XI | NULL | 19 | SSCH | NULL |
+----+----+----+----+----+
1 row in set (0.001 sec)
```

(59) 新增數據到 V_SSCH

7. 通过视图 V_SSCH，删除学号为“S12”学生信息，并查询结果。

```
delete from V_SSCH where Sno="S12";
select * from V_SSCH;
```

這裡我不小心輸錯了，重輸一遍。

```

MariaDB [student_data]> delete from V_SSCH where Sno="S12";
Query OK, 1 row affected (0.009 sec)

MariaDB [student_data]> select * from V_SSCH where;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
corresponds to your MariaDB server version for the right syntax to use
line 1
MariaDB [student_data]> select * from V_SSCH;
+-----+-----+-----+-----+-----+
| Sno   | Sname | Ssex    | Sage  | sdept | SBirthday |
+-----+-----+-----+-----+-----+
| 222222222 | 我     | 不知道  | 22    | SSCH   | NULL      |
| 3333333333 | 好     | 你猜猜  | 13    | SSCH   | NULL      |
| 7777777777 | 菜     | 無       | 30    | SSCH   | NULL      |
| 8888888888 | 阿     | 女       | 8     | SSCH   | NULL      |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

```

(60) 從 V_SSCH 刪除一筆數據

8. 將視圖 V_SSCH 中學號為“S12”的學生改名“中南人”。

```
update V_SSCH set Sno="中南人" where Sno="S12";
```

```

MariaDB [student_data]> update V_SSCH set Sno="中南人" where Sno="S12";
Query OK, 1 row affected (0.008 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [student_data]> select * from V_SSCH;
+-----+-----+-----+-----+-----+
| Sno   | Sname | Ssex    | Sage  | sdept | SBirthday |
+-----+-----+-----+-----+-----+
| 222222222 | 我     | 不知道  | 22    | SSCH   | NULL      |
| 3333333333 | 好     | 你猜猜  | 13    | SSCH   | NULL      |
| 7777777777 | 菜     | 無       | 30    | SSCH   | NULL      |
| 8888888888 | 阿     | 女       | 8     | SSCH   | NULL      |
| 中南人     | YAN XI | NULL    | 19    | SSCH   | NULL      |
+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

```

(61) 從 V_SSCH 更新一筆數據

0.14.2 数据完整性控制

創建約束

1. 定義實體完整性

添加主鍵約束。

發現自己上個實驗就已經設主鍵了。

```
alter table S add primary key(Sno);
```

```
MariaDB [student_data]> alter table S add constraint constraint_s primary key(Sno);
ERROR 1068 (42000): Multiple primary key defined
MariaDB [student_data]>
MariaDB [student_data]>
MariaDB [student_data]> describe S;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Sno   | varchar(20) | NO   | PRI | NULL    |       |
| Sname | varchar(10)  | YES  |     | NULL    |       |
| Ssex  | varchar(5)   | YES  |     | NULL    |       |
| Sage  | int(11)     | YES  |     | NULL    |       |
| sdept | varchar(20)  | YES  |     | NULL    |       |
| SBirthday | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.006 sec)
```

(62) S 設 primary key

試試網上找到的其他寫法，結果報了 Warning，但還是成功了。

```
alter table C add constraint constraint_c primary key(Cno);
```

```

MariaDB [student_data]> alter table C add constraint constraint_c primary key(Cno);
Query OK, 0 rows affected, 1 warning (0.095 sec)
Records: 0  Duplicates: 0  Warnings: 1

MariaDB [student_data]> describe C;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Cno   | varchar(20) | NO   | PRI | NULL    |       |
| Cname | varchar(10)  | YES  |     | NULL    |       |
| Cpno  | varchar(10)  | YES  |     | NULL    |       |
| ccredit | int(11)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

```

(63) C 設 primary key

```
alter table SC add primary key(Sno, Cno);
```

```

MariaDB [student_data]> alter table SC add primary key(Sno, Cno);
Query OK, 0 rows affected (0.075 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [student_data]> describe SC;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Sno   | varchar(20) | NO   | PRI | NULL    |       |
| Cno   | varchar(20) | NO   | PRI | NULL    |       |
| grade | int(11)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

```

(64) SC 設 primary key

2. 參照完整性

設置外鍵：

```
alter table SC add foreign key(Sno) references S(Sno);
```

因為 SC 有些 Sno 不在父 S 的 Sno 裡所以出現下面第一張圖的錯誤，我將 SC 的資料修一下再次執行就好了。

```
MariaDB [student_data]> alter table SC add foreign key(Sno) references S(Sno);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (
`student_data`.`#sql-264_8`, CONSTRAINT `#sql-264_8_ibfk_1` FOREIGN KEY (`Sno`) REFERER-
NCES `S` (`Sno`))
```

(65) SC 設 foreign key 失敗

```
MariaDB [student_data]> alter table SC add foreign key(Sno) references S(Sno);
Query OK, 3 rows affected (0.087 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

(66) SC 設 foreign key

3. 用戶定義完整性

```
alter table S add check(Sage < 30);
```

修改約束條件

```
Alter table S add check(Sno between 100 and 1000);
```

0.15 實驗小結

這次的實驗讓我學會怎麼給數據庫添加 View，還有主鍵和外鍵，最後也讓我時做了約束條件。我覺得真的有學到的感覺，也對數據庫又更深一步的了解。

Part IV

實驗四、存儲過程

0.16 實驗目的

理解存储过程的概念、建立和调用方法。进一步熟悉 SQL 语句对数据库进行完整性控制的方法。

0.17 實驗內容

- 利用存储过程查找自己的学号、选修课程及成绩。
- 利用存储过程查找姓“李”并且性别为“M”的学生学号、选修课程。
- 利用存储过程计算某同学（学号作为存储过程的参数）所选课程的平均分。

0.18 實驗方法與實驗步驟

- 设计查找自己的学号、选修课程及成绩的存储过程，以自己的学号作为参数，调用存储过程。
- 设计存储过程查找姓“李”并且性别为“M”的学生学号、选修课程并调用。
- 设计存储过程计算某同学（学号作为存储过程的参数）所选课程的平均分并调用。

0.19 實驗結果

1. 设计查找自己的学号、选修课程及成绩的存储过程，以自己的学号作为参数，调用存储过程。

建立 Procedure。

```
delimiter //
create procedure studentSearch(in inputSno int)
begin
```

```
select S.sno, S.sname, SC.cno from S, SC where inputSno=S.sno and SC.sno=
      inputSno;
END//  
delimiter;
```

設定參數並運行。

```
set @inputSno = 1;  
call studentSearch(@inputSno);
```

2. 设计存储过程查找姓“李”并且性别为“M”的学号、选修课程并调用。

```
delimiter //  
create procedure search_lee()  
begin  
select S.sno, SC.cno, SC.grade from S, SC where S.sname Like '李%';  
end//  
delimiter;  
  
call search_lee();
```

3. 设计存储过程计算某同学（学号作为存储过程的参数）所选课程的平均分并调用。

建立 Procedure。

```
delimiter //  
create procedure search_avg(in inputSno int)  
begin  
select S.sno,avg(SC.grade) from S,SC where S.sno=inputSno and SC.sno=inputSno  
;  
end//  
delimiter;
```

設定參數並運行。

```
SET @inputSno=1;
call searchavg(@inputSno);
```

0.20 實驗小結

因為我是使用 MySQL 做實驗所以這次的實驗一開始使用了 SQL 的語法，結果發現運行錯誤，到網上搜索了一下才發現 Store Procedure 上 MySQL 與 SQL 不一樣，最後改成了正確的 MySQL 語法。

Part V

實驗五、觸發器游標

0.21 實驗目的

进一步熟悉 SQL 语句对数据库进行完整性控制的方法；理解触发器的概念、定义方法和触发条件。理解游标的定义、打开、使用、关闭与释放的方法。

0.22 實驗內容

- 设置一个触发器，该触发器仅允许“dbo”用户可以删除学生表内数据。
- 针对学生表写一个 DELETE 触发器。
- 针对学生表写一个 UPDATE 触发器。
- 统计学生的平均成绩，输出低于平均分的成绩（使用游标）。

0.23 實驗方法與實驗步驟

- 设置一个触发器，该触发器仅允许“dbo”用户可以删除学生表内数据，否则出错。
- 针对学生表写一个 DELETE 触发器，删除数据，测试触发。
- 针对学生表写一个 UPDATE 触发器，更新数据，测试触发。
- 定义游标，统计学生的平均成绩，输出低于平均分的成绩。

0.24 實驗結果

1. 设置一个触发器，该触发器仅允许“dbo”用户可以删除学生表内数据，否则出错。

```
create trigger dbo_delete
on S
for delete
```

```
as
if exists (select * from deleted)
begin
    if USER != 'dbo'
end
delete from S where S.Sno = '1'
```

2. 针对学生表写一个 DELETE 触发器，删除数据，测试触发。

```
create trigger S_delete
on S
for delete
as
print('DELETE')
delete from S where S.Sno = '1'
```

3. 针对学生表写一个 UPDATE 触发器，更新数据，测试触发。

```
create trigger S_update
on S
for update
as
print(update)
update S set S.sdept = 'CS' where S.Sno = '1'
```

4. 定义游标，统计学生的平均成绩，输出低于平均分的成绩。

```
declare lowAvg cursor
for
select SC.grade
from SC
where SC.grade < (select avg(SC.grade) from SC)

open lowAvg
fetch next
from S
into @grade

while @fetch_status=0
begin
print(@grade)
fetch next from lowAvg into @grade
```

```
end  
close lowAvg  
deallocate lowAvg
```

0.25 實驗小結

這次的實驗繞我學到了游標和觸發器，我也在網上看了關於這方面的語法，觸發器是為了觸發某個動作的機制，游標是在存儲過程的瀏覽。因為之前寫項目都沒接觸到這部份，這次讓我親自體驗這之中的意思。果然親自動手做實驗能讓人對這方面的知識更清晰。