

《数据库技术实验》报告

第 二 次实验	日期：2022-09-29	得分：
学号：	姓名：Steven	专业：智能科学与技术

一、实验目的

1. 掌握插入、更新、删除数据的 SQL 语句。
2. 掌握简单查询数据的 SQL 语句。

二、实验内容

(一) 实验题

1. 删除 Student_Course 数据库中 学生表 Student、课程表 Course、学生选课表 SC 里的所有记录。
2. 在 Student_Course 数据库中 学生表 Student、课程表 Course、学生选课表 SC 中插入以下记录。
3. 计算机系学生的年龄减 1
4. 数据库课程的成绩加 5
5. 将计算机系学生的选课信息存入表 SC_cs 中
6. 删除学生张三的选课信息
7. 将 World 数据库中 country 表中的国家信息 Code、Name、Capital 存入表 Country_Simple 中
8. 将 World 数据库中 city 表中的城市信息 ID、Name、CountryCode 存入表 City_Simple 中
9. 将 World 数据库中 countrylanguage 表中的国家语言信息 CountryCode、Language 存入表 CL_Simple 中

(二) 思考题

1. 在表中插入记录时，如果已存在主码相同的记录，DBMS 如何处理？如果插入记录的某些值没有指定，DBMS 如何处理？如果插入记录有外码，DBMS 如何处理？
2. 从表中删除记录时，如果删除记录的主码是其他表记录的外码，DBMS 如何处理？
3. 修改表中记录时，如果修改主码，DBMS 如何处理？

三、实验过程及实验结果

1.

由于一开始是没有数据的，所以先创建一些测试项并查询：

<pre>use student_course; INSERT INTO student SET sno='20201060351', sname='Zh1', ssex='男', sage=20, sdept='智能科学与技术'; INSERT INTO course SET cno='cno_数据库', cname='数据库技术', cpno=NULL, ccredit=3; INSERT INTO sc SET</pre>	<pre>mysql> SELECT * FROM student; +-----+-----+-----+-----+-----+ sno sname ssex sage sdept +-----+-----+-----+-----+-----+ 20201060351 Zh1 男 20 智能科学与技术 +-----+-----+-----+-----+-----+ 1 row in set (0.00 sec) mysql> SELECT * FROM course; +-----+-----+-----+-----+-----+ cno cname cpno ccredit +-----+-----+-----+-----+-----+ cno_数据库 数据库技术 NULL 3 +-----+-----+-----+-----+-----+</pre>
--	--

sno='20201060351', cno='cno_数据库', grade=3;	1 row in set (0.00 sec) mysql> SELECT * FROM sc; +-----+-----+-----+ sno cno grade +-----+-----+-----+ 20201060351 cno_数据库 3 +-----+-----+-----+ 1 row in set (0.00 sec)
--	---

随后删除:

use student_course; DELETE FROM sc; DELETE FROM student; DELETE FROM course;	mysql> SELECT * FROM student; Empty set (0.00 sec) mysql> SELECT * FROM course; Empty set (0.00 sec) mysql> SELECT * FROM sc; Empty set (0.00 sec)
---	---

表示所有数据都已删除。

2.

按照题目要求录入信息:

use student_course; INSERT INTO student SET sno='20191060001',sname='张三',ssex='男',sage=20,sdept='计算机'; INSERT INTO student SET sno='20191060002',sname='李四',ssex='女',sage=19,sdept='通信工程'; INSERT INTO student SET sno='20191060003',sname='王五',ssex='男',sage=20,sdept='计算机'; INSERT INTO student SET sno='20191060004',sname='赵六',ssex='男',sage=19,sdept='通信工程'; INSERT INTO student SET sno='20191060005',sname='钱七',ssex='女',sage=18,sdept='计算机'; INSERT INTO course SET cno='1',cname='C 语言',cpno=NULL,ccredit=3; INSERT INTO course SET cno='2',cname='数据结构',cpno='1',ccredit=3; INSERT INTO course SET cno='3',cname='数据库',cpno='2',ccredit=4; INSERT INTO course SET cno='4',cname='操作系统',cpno='2',ccredit=4; INSERT INTO course SET cno='5',cname='数据挖掘',cpno='3',ccredit=2; INSERT INTO course SET cno='6',cname='人工智能',cpno='2',ccredit=2; INSERT INTO sc SET sno='20191060001',cno='1',grade=68; INSERT INTO sc SET sno='20191060001',cno='2',grade=76; INSERT INTO sc SET sno='20191060001',cno='3',grade=80; INSERT INTO sc SET sno='20191060002',cno='1',grade=77; INSERT INTO sc SET sno='20191060002',cno='2',grade=80; INSERT INTO sc SET sno='20191060002',cno='4',grade=85; INSERT INTO sc SET sno='20191060003',cno='1',grade=90; INSERT INTO sc SET sno='20191060003',cno='2',grade=92; INSERT INTO sc SET sno='20191060003',cno='3',grade=90; INSERT INTO sc SET sno='20191060003',cno='5',grade=91; INSERT INTO sc SET sno='20191060004',cno='1',grade=88; INSERT INTO sc SET sno='20191060004',cno='6',grade=90; INSERT INTO sc SET sno='20191060005',cno='1',grade=86; INSERT INTO sc SET sno='20191060005',cno='5',grade=88;

查询结果如下:

```
mysql> SELECT * FROM student;
```

sno	sname	ssex	sage	sdept
20191060001	张三	男	20	计算机
20191060002	李四	女	19	通信工程
20191060003	王五	男	20	计算机
20191060004	赵六	男	19	通信工程
20191060005	钱七	女	18	计算机

5 rows in set (0.00 sec)

```
mysql> SELECT * FROM course;
```

cno	cname	cpno	ccredit
1	C 语言	NULL	3
2	数据结构	1	3
3	数据库	2	4
4	操作系统	2	4
5	数据挖掘	3	2
6	人工智能	2	2

6 rows in set (0.00 sec)

```
mysql> SELECT * FROM sc;
```

sno	cno	grade
20191060001	1	68
20191060001	2	76
20191060001	3	80
20191060002	1	77
20191060002	2	80
20191060002	4	85
20191060003	1	90
20191060003	2	92
20191060003	3	90
20191060003	5	91
20191060004	1	88
20191060004	6	90
20191060005	1	86
20191060005	5	88

14 rows in set (0.00 sec)

证明插入数据完成。

3.

先查看一下最初的数据：

```
mysql> SELECT * FROM student;
```

sno	sname	ssex	sage	sdept
20191060001	张三	男	20	计算机
20191060002	李四	女	19	通信工程
20191060003	王五	男	20	计算机
20191060004	赵六	男	19	通信工程
20191060005	钱七	女	18	计算机

5 rows in set (0.00 sec)

随后更新数据，使用 UPDATE 指令：

```
UPDATE student SET sage = sage - 1 WHERE sdept='计算机';
```

再次查询，检查是否完成修改：

```
mysql> SELECT * FROM student;
```

sno	sname	ssex	sage	sdept
20191060001	张三	男	19	计算机
20191060002	李四	女	19	通信工程
20191060003	王五	男	19	计算机
20191060004	赵六	男	19	通信工程
20191060005	钱七	女	17	计算机

5 rows in set (0.00 sec)

发现年龄已经自减 1，完成修改。

4.

先查看一下 sc，发现只有如下标出的学生选了数据库课程：

```
mysql> SELECT * FROM sc;
```

sno	cno	grade
20191060001	1	68
20191060002	2	80
20191060002	4	85
20191060003	1	90
20191060003	2	92
20191060003	3	90
20191060003	5	91
20191060004	1	88
20191060004	6	90
20191060005	1	86
20191060005	5	88

14 rows in set (0.00 sec)

随后进行修改并再次查看，发现数据库的成绩已经加了 5 分：

```
UPDATE sc
SET grade = grade + 5
WHERE cno = (
    SELECT cno FROM course
    WHERE cname = '数据库'
);
```

```
mysql> SELECT * FROM sc;
```

sno	cno	grade
20191060001	1	68
20191060001	2	76
20191060001	3	85
20191060002	1	77
20191060002	2	80
20191060002	4	85
20191060003	1	90
20191060003	2	92
20191060003	3	95
20191060003	5	91
20191060004	1	88
20191060004	6	90
20191060005	1	86
20191060005	5	88

```
14 rows in set (0.00 sec)
```

5.

首先创建一个表 SC_cs:

```
CREATE TABLE SC_cs(
    sno CHAR(11) NOT NULL,
    cno CHAR(11) NOT NULL,
    grade SMALLINT,
    PRIMARY KEY(sno, cno)
);
```

由于需要通过 student.sdept 确定学号，于是这个问题需要用到两张表，在此过程中意外发现了两种解决方式：

① 第一种：先找出所有计算机系的学号，再将其在 SC 中进行检查，将存在所有学号中的学生的选课记录拷贝。

```
INSERT INTO sc_cs
SELECT SC.* FROM SC WHERE sno IN (SELECT sno FROM student WHERE sdept = '计算机');
```

② 第二种：将 SC 和 student 根据 sno 进行连接，然后直接在其中选择 sdept 为计算机的选课信息，然后进行拷贝。

```
INSERT INTO SC_cs
SELECT SC.* FROM SC INNER JOIN student ON SC.sno = student.sno WHERE sdept = '计算机';
```

然后查询结果，均可以完成需求：

```
mysql> SELECT * FROM sc_cs;
```

sno	cno	grade
20191060001	1	68
20191060001	2	76
20191060001	3	85
20191060003	1	90
20191060003	2	92

20191060003	3	95
20191060003	5	91
20191060005	1	86
20191060005	5	88

9 rows in set (0.00 sec)

这其中包括 20191060001、20191060003、20191060005 三名同学的选课信息，经查这三名同学都是计算机系的学生，故说明实验成功。

但是通常来说，连接操作的性能的开支很大，第一种方法应该更好。

6.

先进行一下查询，随后进行删除操作，其逻辑为从 student 中获得张三的学号，然后根据这一点的等式约束将其从 sc 中删除，最后再检查一下，发现张三（20191060001）的选课数据已被删除。

mysql> SELECT * FROM sc;	DELETE FROM sc	mysql> SELECT * FROM sc;																																																																																	
<table border="1"> <tr><th>sno</th><th>cno</th><th>grade</th></tr> <tr><td>20191060001</td><td>1</td><td>68</td></tr> <tr><td>20191060001</td><td>2</td><td>76</td></tr> <tr><td>20191060001</td><td>3</td><td>85</td></tr> <tr><td>20191060002</td><td>1</td><td>77</td></tr> <tr><td>20191060002</td><td>2</td><td>80</td></tr> <tr><td>20191060002</td><td>4</td><td>85</td></tr> <tr><td>20191060003</td><td>1</td><td>90</td></tr> <tr><td>20191060003</td><td>2</td><td>92</td></tr> <tr><td>20191060003</td><td>3</td><td>95</td></tr> <tr><td>20191060003</td><td>5</td><td>91</td></tr> <tr><td>20191060004</td><td>1</td><td>88</td></tr> <tr><td>20191060004</td><td>6</td><td>90</td></tr> <tr><td>20191060005</td><td>1</td><td>86</td></tr> <tr><td>20191060005</td><td>5</td><td>88</td></tr> </table>	sno	cno	grade	20191060001	1	68	20191060001	2	76	20191060001	3	85	20191060002	1	77	20191060002	2	80	20191060002	4	85	20191060003	1	90	20191060003	2	92	20191060003	3	95	20191060003	5	91	20191060004	1	88	20191060004	6	90	20191060005	1	86	20191060005	5	88	WHERE sno=(SELECT sno FROM student WHERE sname='张三');	<table border="1"> <tr><th>sno</th><th>cno</th><th>grade</th></tr> <tr><td>20191060002</td><td>1</td><td>77</td></tr> <tr><td>20191060002</td><td>2</td><td>80</td></tr> <tr><td>20191060002</td><td>4</td><td>85</td></tr> <tr><td>20191060003</td><td>1</td><td>90</td></tr> <tr><td>20191060003</td><td>2</td><td>92</td></tr> <tr><td>20191060003</td><td>3</td><td>95</td></tr> <tr><td>20191060003</td><td>5</td><td>91</td></tr> <tr><td>20191060004</td><td>1</td><td>88</td></tr> <tr><td>20191060004</td><td>6</td><td>90</td></tr> <tr><td>20191060005</td><td>1</td><td>86</td></tr> <tr><td>20191060005</td><td>5</td><td>88</td></tr> </table>	sno	cno	grade	20191060002	1	77	20191060002	2	80	20191060002	4	85	20191060003	1	90	20191060003	2	92	20191060003	3	95	20191060003	5	91	20191060004	1	88	20191060004	6	90	20191060005	1	86	20191060005	5	88
sno	cno	grade																																																																																	
20191060001	1	68																																																																																	
20191060001	2	76																																																																																	
20191060001	3	85																																																																																	
20191060002	1	77																																																																																	
20191060002	2	80																																																																																	
20191060002	4	85																																																																																	
20191060003	1	90																																																																																	
20191060003	2	92																																																																																	
20191060003	3	95																																																																																	
20191060003	5	91																																																																																	
20191060004	1	88																																																																																	
20191060004	6	90																																																																																	
20191060005	1	86																																																																																	
20191060005	5	88																																																																																	
sno	cno	grade																																																																																	
20191060002	1	77																																																																																	
20191060002	2	80																																																																																	
20191060002	4	85																																																																																	
20191060003	1	90																																																																																	
20191060003	2	92																																																																																	
20191060003	3	95																																																																																	
20191060003	5	91																																																																																	
20191060004	1	88																																																																																	
20191060004	6	90																																																																																	
20191060005	1	86																																																																																	
20191060005	5	88																																																																																	
14 rows in set (0.00 sec)		11 rows in set (0.00 sec)																																																																																	

检查一下结果：发现张三（20191060001）的选课记录已被删除

7.

7-9 这三个题的思路都相同，均是创建一个包含部分属性的新的表，然后按顺序从原表中拷贝信息即可。

```
use world;

CREATE TABLE
  Country_Simple(
    Code CHAR(3) NOT NULL,
    Name CHAR(52) NOT NULL,
    Capital INT DEFAULT NULL,
    PRIMARY KEY(Code)
  );
```

```
INSERT INTO Country_Simple SELECT Code,Name,Capital FROM country;
```

```
mysql> SELECT * FROM Country_Simple LIMIT 50;
```

Code	Name	Capital
ABW	Aruba	129
AFG	Afghanistan	1
AGO	Angola	56
AIA	Anguilla	62
ALB	Albania	34
AND	Andorra	55
ANT	Netherlands Antilles	33
ARE	United Arab Emirates	65
ARG	Argentina	69
ARM	Armenia	126
ASM	American Samoa	54
ATA	Antarctica	NULL
ATF	French Southern territories	NULL
ATG	Antigua and Barbuda	63
AUS	Australia	135
AUT	Austria	1523
AZE	Azerbaijan	144
BDI	Burundi	552
BEL	Belgium	179
BEN	Benin	187
BFA	Burkina Faso	549
BGD	Bangladesh	150
BGR	Bulgaria	539
BHR	Bahrain	149
BHS	Bahamas	148
BIH	Bosnia and Herzegovina	201
BLR	Belarus	3520
BLZ	Belize	185
BMU	Bermuda	191
BOL	Bolivia	194
BRA	Brazil	211
BRB	Barbados	174
BRN	Brunei	538
BTN	Bhutan	192
BVT	Bouvet Island	NULL
BWA	Botswana	204
CAF	Central African Republic	1889
CAN	Canada	1822
CCK	Cocos (Keeling) Islands	2317
CHE	Switzerland	3248
CHL	Chile	554

CHN	China	1891
CIV	Côte d' Ivoire	2814
CMR	Cameroon	1804
COD	Congo, The Democratic Republic of the	2298
COG	Congo	2296
COK	Cook Islands	583
COL	Colombia	2257
COM	Comoros	2295
CPV	Cape Verde	1859

50 rows in set (0.00 sec)

8.

```
use world;
```

```
CREATE TABLE
```

```
City_Simple(
  ID INT NOT NULL AUTO_INCREMENT,
  Name CHAR(35) NOT NULL,
  CountryCode CHAR(3) NOT NULL,
  PRIMARY KEY (ID)
);
```

```
INSERT INTO City_Simple SELECT ID,Name,CountryCode FROM city;
```

```
mysql> SELECT * FROM City_Simple LIMIT 50;
```

ID	Name	CountryCode
1	Kabul	AFG
2	Qandahar	AFG
3	Herat	AFG
4	Mazar-e-Sharif	AFG
5	Amsterdam	NLD
6	Rotterdam	NLD
7	Haag	NLD
8	Utrecht	NLD
9	Eindhoven	NLD
10	Tilburg	NLD
11	Groningen	NLD
12	Breda	NLD
13	Apeldoorn	NLD
14	Nijmegen	NLD
15	Enschede	NLD
16	Haarlem	NLD
17	Almere	NLD

18	Arnhem	NLD	
19	Zaanstad	NLD	
20	's-Hertogenbosch	NLD	
21	Amersfoort	NLD	
22	Maastricht	NLD	
23	Dordrecht	NLD	
24	Leiden	NLD	
25	Haarlemmermeer	NLD	
26	Zoetermeer	NLD	
27	Emmen	NLD	
28	Zwolle	NLD	
29	Ede	NLD	
30	Delft	NLD	
31	Heerlen	NLD	
32	Alkmaar	NLD	
33	Willemstad	ANT	
34	Tirana	ALB	
35	Alger	DZA	
36	Oran	DZA	
37	Constantine	DZA	
38	Annaba	DZA	
39	Batna	DZA	
40	Sétif	DZA	
41	Sidi Bel Abbès	DZA	
42	Skikda	DZA	
43	Biskra	DZA	
44	Blida (el-Boulaida)	DZA	
45	Béjaïa	DZA	
46	Mostaganem	DZA	
47	Tébessa	DZA	
48	Tlemcen (Tilimsen)	DZA	
49	Béchar	DZA	
50	Tiaret	DZA	

+-----+-----+-----+

50 rows in set (0.00 sec)

9.

```
use world;
```

```
CREATE TABLE
```

```
  CL_Simple(
    CountryCode CHAR(3) NOT NULL,
    Language CHAR(30) NOT NULL,
    PRIMARY KEY (CountryCode, Language)
  );
```

```
INSERT INTO CL_Simple
```

```
SELECT CountryCode, Language
FROM countrylanguage;
```

```
mysql> SELECT * FROM CL_Simple LIMIT 50;
```

CountryCode	Language
ABW	Dutch
ABW	English
ABW	Papiamentu
ABW	Spanish
AFG	Balochi
AFG	Dari
AFG	Pashto
AFG	Turkmenian
AFG	Uzbek
AGO	Ambo
AGO	Chokwe
AGO	Kongo
AGO	Luchazi
AGO	Luimbe-nganguela
AGO	Luvale
AGO	Mbundu
AGO	Nyaneka-nkhumbi
AGO	Ovimbundu
AIA	English
ALB	Albaniana
ALB	Greek
ALB	Macedonian
AND	Catalan
AND	French
AND	Portuguese
AND	Spanish
ANT	Dutch
ANT	English
ANT	Papiamentu
ARE	Arabic
ARE	Hindi
ARG	Indian Languages
ARG	Italian
ARG	Spanish
ARM	Armenian
ARM	Azerbaijani
ASM	English
ASM	Samoan
ASM	Tongan
ATG	Creole English

ATG	English	
AUS	Arabic	
AUS	Canton Chinese	
AUS	English	
AUS	German	
AUS	Greek	
AUS	Italian	
AUS	Serbo-Croatian	
AUS	Vietnamese	
AUT	Czech	
+-----+-----+		
50 rows in set (0.00 sec)		

四、思考题

1.

- ① 因为主码是唯一标识一个元组的，所有出现主码重复时，应当是无法插入的。事实也确实如此：

```
mysql> SELECT * FROM course;
+----+-----+-----+-----+
| cno | cname      | cpno | ccredit |
+----+-----+-----+-----+
| 1   | C 语言      | NULL | 3       |
| 2   | 数据结构    | 1    | 3       |
| 3   | 数据库      | 2    | 4       |
| 4   | 操作系统    | 2    | 4       |
| 5   | 数据挖掘    | 3    | 2       |
| 6   | 人工智能    | 2    | 2       |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> INSERT INTO course (cno,cname,ccredit) VALUES ('2','test',3);
ERROR 1062 (23000): Duplicate entry '2' for key 'course.PRIMARY'
```

上述操作试图插入一个 cno 为 2 的课，但是这与已有的数据结构冲突，因而无法插入。

- ② 这一点要根据缺少的值来决定。通常来说，若缺少主码，则无法插入，但是若缺少的值设置了缺省值 Default，则会填入 Default 值。

```
mysql> DESC sc;
+----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| sno   | char(11)  | NO   | PRI | NULL    |       |
| cno   | char(11)  | NO   | PRI | NULL    |       |
| grade | smallint  | YES  |     | NULL    |       |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> INSERT INTO sc (cno,grade) VALUES ('2',85);
ERROR 1364 (HY000): Field 'sno' doesn't have a default value
```

```
mysql> INSERT INTO sc (sno,cno) VALUES ('20191060002','5');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM sc WHERE sno='20191060002' AND cno='5';
+-----+-----+-----+
| sno          | cno | grade |
+-----+-----+-----+
| 20191060002 | 5   | NULL  |
+-----+-----+-----+
1 row in set (0.00 sec)
```

上图所示，sc 表中两个主码都不可为 NULL，而 grade 可为 NULL 且 Default 为 NULL。于是第一次插入数据时由于没有填 sno 参数而无法填入；而第二次在指定 sno 和 cno 后即可插入，其 grade 值为 NULL 即为缺省值。

- ③ 插入存在外码，则会执行参照完整性检验，存在的值允许插入，不存在的不允许。

```
mysql> INSERT INTO sc SET sno='test_stu', cno='5';
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`student_course`.`sc`, CONSTRAINT `sc_ibfk_1` FOREIGN KEY (`sno`) REFERENCES `student` (`sno`))
```

例如这个，插入时提示无法插入，因为有 student.sno 作为参照的约束。

2. 这句话有点绕，意思是有一个表参照了该表的主码，那么在删除该表的元组时能否删除。

其实是不能直接删的，因为存在参照完整性检验。所以真的想删除的话，要先把参照完整性检验关掉再删。当然最好是开着，不然外码就失去其价值了。

```
mysql> DELETE FROM student WHERE sname='钱七';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails
(`student_course`.`sc`, CONSTRAINT `sc_ibfk_1` FOREIGN KEY (`sno`) REFERENCES `student` (`sno`))

mysql> SET foreign_key_checks = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM student WHERE sname='钱七';
Query OK, 1 row affected (0.00 sec)

mysql> SET foreign_key_checks = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM student;
+-----+-----+-----+-----+-----+
| sno          | sname | ssex | sage | sdept |
+-----+-----+-----+-----+-----+
| 20191060001 | 张三  | 男   | 19   | 计算机 |
| 20191060002 | 李四  | 女   | 19   | 通信工程 |
| 20191060003 | 王五  | 男   | 19   | 计算机 |
| 20191060004 | 赵六  | 男   | 19   | 通信工程 |
+-----+-----+-----+-----+-----+
```

上述操作中删除钱七没成功，因为存在参照完整性检验。随后关闭了参照完整性检验，成功删除。

- 3.

在满足实体完整性的情况下，是可以直接改的。

```
mysql> UPDATE student SET sno = 'test_sno' WHERE sno = '20191060001';
```

```
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE student SET sno = '20191060002' WHERE sno = 'test_sno';
ERROR 1062 (23000): Duplicate entry '20191060002' for key 'student.PRIMARY'
```

上述操作中，将 sno 修改为'test_sno'，是可以的，但是将其修改为'20191060002'时，由于这个主码已经存在，所以不能修改。