

# 数据库实验报告 实验十 数据库完整性

姓名	学号	班级	课室
熊明	20305055	计科5班	D503

## 一、实验目的

学习实体完整性的建立，以及实践违反实体完整性的结果；学习建立外键，以及利用 FOREIGN KEY... REFERENCES子句以及各种约束保证参照完整性。

## 二、实验环境

数据库：Mysql

图形化工具：Navicat Premium 16

## 三、实验内容

1. 实体完整性：保证关系中的每个元组都是可识别的和惟一的。关系数据库中所有的表都必须有主键，而且表中不允许存在如下记录：无主键值的记录，主键值相同的记录。因为实体必须可区分
2. 参考完整性：现实世界中的实体之间往往存在着某种联系，在关系模型中，实体以及实体之间的联系都是用关系来表示的，这样就自然存在着关系与关系之间的引用
  - 参照完整性就是描述实体之间的联系的
  - 参照完整性一般是指多个实体或关系之间的关联关系
- 3.

## 四、课内实验

1. 在数据库 school中建立表Stu\_Union, 进行主键约束，在没有违反实体完整性的前提下插入并更新一条记录。(参考代码如下：)

```
1 CREATE TABLE Stu_Union(  
2     sno CHAR(5) NOT NULL UNIQUE,  
3     sname CHAR(8),  
4     ssex CHAR(1),  
5     sage INT,  
6     sdept CHAR(20),  
7     CONSTRAINT PK_Stu_Union PRIMARY KEY(sno)  
8 );  
9  
10 insert Stu_Union values('10000','王敏','1',23,'cs');  
11  
12 UPDATE Stu_Union SET sno='' WHERE sdept='CS';  
13 UPDATE Stu_Union SET sno='95002' WHERE sname='王敏';  
14  
15 select * from Stu_Union;
```

运行结果如下：

信息	摘要	结果 1	剖析	状态	
	sno	sname	ssex	sage	sdept
▶	95002	王敏	1	23	cs

2. 演示违反实体完整性的插入操作。

执行下面语句：

```
1 INSERT Stu_Union VALUES(null,'xm','1',21,'cs');
```

得到如下报错：

信息

1048 - Column 'sno' cannot be null

3. 演示违反实体完整性的更新操作。

先创建一个记录：

```
1 INSERT Stu_Union VALUES('10001','xm','1',21,'cs');
```

然后利用如下语句更新，尝试将xm的sno改为95002：

```
1 UPDATE Stu_Union SET sno='95002' WHERE sname = 'xm'
```

得到如下结果：

信息

1062 - Duplicate entry '95002' for key 'stu\_union.PRIMARY'

4. 为演示参照完整性，建立表 Course,令 cno 为其主键，并在 Stu\_Union 中插入数据。为下面的实验步骤做预先准备。(参考代码如下：)

```
1 insert into Stu_Union values('10001','李勇','0',24,'EE');
2
3 select * from Stu_Union;
4
5 create table Course(
6     cno char(4)NOT NULL UNIQUE,
7     cname varchar(50)NOT NULL,
8     cpoints int,
9     constraint PK primary key(cno)
10 );
11
12 insert into Course values('0001','ComputerNetworks',2);
13 insert into Course values('0002','Databae',3);
14 SELECT * FROM course;
```

结果如下：

sno	sname	ssex	sage	sdept
▶ 10001	李勇	0	24	EE
95002	王敏	1	23	CS

cno	cname	cpoints
▶ 0001	ComputerNetworks	2
0002	Databases	3

5. 建立表 SC, 令 sno 和 cno 分别为参照 stu union 表以及 Course 表的外键, 设定为 级联删除, 并令(sno,cno) 为其主键。在不违反参照完整性的前提下, 插入数据。(参考代码如下: )

实验网站上的参考代码没有将sno和cno添加进外键, 只设置成了主键, 会导致下面7、8题得不到正确的结果, 这里做了修改, 将sno和cno添加成外键。

```

1 CREATE TABLE SC(
2     Sno CHAR(5),
3     FOREIGN KEY(sno) REFERENCES Stu_Union(sno) ON DELETE CASCADE,
4     Cno CHAR(4),
5     FOREIGN KEY(cno) REFERENCES Course(cno) ON DELETE CASCADE,
6     grade INT,
7     CONSTRAINT PK_SC PRIMARY KEY(sno,cno)
8 );
9
10 SET FOREIGN_KEY_CHECKS = 0;      # 临时取消外键约束
11 insert into sc values('95002','0001',2);
12 insert into sc values('95002','0002',2);
13 insert into sc values('10001','0001',2);
14 insert into sc values('10001','0002',2);
15 SET FOREIGN_KEY_CHECKS = 1;      # 恢复外键约束
16
17 select * From SC;

```

得到结果如下:

Sno	Cno	grade
▶ 10001	0001	2
10001	0002	2
95002	0001	2
95002	0002	2

6. 演示违反参照完整性的插入数据

执行下面语句:

```

1 insert into sc values(NULL , '0002', 2);

```

结果如下:

信息

1048 - Column 'Sno' cannot be null

7. 在 Stu\_Union 中删除数据，演示级联删除。

原始状态sc表如下：

Sno	Cno	grade
10001	0001	2
10001	0002	2
95002	0001	2
95002	0002	2

删除Stu\_Union表中sno=95002的记录：

```
1 DELETE FROM Stu_Union WHERE sno = '95002';
```

sc表如下：

信息	摘要	结果 1	剖析	状态
Sno	Cno	grade		
10001	0001	2		
10001	0002	2		

8. 在Course 中删除数据，演示级联删除

接着上一题，执行下面语句：

```
1 DELETE FROM course WHERE cno = '0001';
```

sc表如下：

Sno	Cno	grade
10001	0002	2

## 五、自我实践

1. 用 alter table 语句将SC 表中的 on delete cascade 改为 ,重新插入SC 的数据。重复课内实验中7.和8.,观察结果，分析原因。

使用如下语句：

发现无法删除外键约束，报错显示没有sno和cno的约束名。发现定义约束时没有设置约束名，可以利用 `SHOW CREATE TABLE sc;` 展示默认约束名，如下：

```

1 CREATE TABLE `sc` (
2   `Sno` char(5) NOT NULL,
3   `Cno` char(4) NOT NULL,
4   `grade` int DEFAULT NULL,
5   PRIMARY KEY (`Sno`, `Cno`),
6   KEY `Cno` (`Cno`),
7   CONSTRAINT `sc_ibfk_1` FOREIGN KEY (`Sno`) REFERENCES `stu_union`
  (`sno`) ON DELETE CASCADE,
8   CONSTRAINT `sc_ibfk_2` FOREIGN KEY (`Cno`) REFERENCES `course` (`cno`)
  ON DELETE CASCADE
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

可以看到默认约束名为 sc\_ibfk\_1 和 sc\_ibfk\_2，然后利用下面语句进行删除和重新添加约束：

```

1 ALTER TABLE SC
2 DROP FOREIGN KEY sc_ibfk_1,
3 DROP FOREIGN KEY sc_ibfk_2;
4 ALTER TABLE SC
5 ADD CONSTRAINT sno_na FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON
  DELETE NO ACTION,
6 ADD CONSTRAINT cnp_na FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE
  NO ACTION;

```

之后重复课内实验7和8，均得到以下报错：

#### 信息

1451 - Cannot delete or update a parent row: a foreign key constraint fails  
 (`school`.`sc`, CONSTRAINT `sno\_na` FOREIGN KEY (`Sno`) REFERENCES  
 `stu\_union` (`sno`))

当取值为No Action或者Restrict时，则当在父表（即外键的来源表）中删除对应记录时，首先检查该记录是否有对应外键，如果有则不允许删除。

2. 使用 alter table 语句将 SC 表中的 on delete cascade 改为 on delete set NULL,重新插入 SC 的数据。重复课内实验中7.和8.,观察结果，分析原因。

删除外键约束，添加新约束：

```

1 ALTER TABLE SC
2 DROP FOREIGN KEY sno_na,
3 DROP FOREIGN KEY cnp_na;
4
5 ALTER TABLE SC
6 DROP PRIMARY KEY;
7
8 ALTER TABLE SC
9 MODIFY COLUMN sno CHAR(5) NULL,
10 MODIFY COLUMN cno CHAR(4) NULL;
11
12 ALTER TABLE SC
13 ADD CONSTRAINT sno_sn FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON
  DELETE set NULL,
14 ADD CONSTRAINT cno_sn FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE
  SET NULL;

```

首先因为之前设置了 sno 和 cno 为主键，逐渐不能为空。所以将 sc 的主键约束删除，然后添加新的约束。

执行以下语句：

```
1 DELETE FROM Stu_Union WHERE sno = '95002';
2 SELECT * FROM sc;
```

得到如下结果：

信息	摘要	结果 1	剖析	状态
	sno	cno	grade	
▶	10001	0001		2
	10001	0002		2
	(Null)	0001		2
	(Null)	0002		2

执行以下语句：

```
1 DELETE FROM course WHERE cno = '0001';
```

得到以下结果：

信息	摘要	结果 1	剖析	状态
	sno	cno	grade	
▶	10001	(Null)		2
	10001	0002		2
	(Null)	(Null)		2
	(Null)	0002		2

当取值为Set Null时，则当在父表（即外键的来源表）中删除对应记录时，首先检查该记录是否有对应外键，如果有则设置子表中该外键值为null（要求该外键允许取null）