



本科生实验报告

学生姓名： 丁晓琪

学生学号： 22336057

专业名称： 计科

1.实验要求

(1)用alter table语句将SC表中的on delete cascade改为on delete no action,重新插入SC的数据（按照实验一）。再删除Stu_Union中sno为'10001'的数据。观察结果，并分析原因。

(2)用alter table语句将SC表中的on delete no action改为on delete set NULL,重新插入SC的数据（按照实验一）。再删除Stu_Union中sno为'10001'的数据。观察结果，并分析原因。

(3)建立事务T3，修改ICBC_Card表的外键属性，使其变为on delete set NULL,尝试删除students表中一条记录。观察结果，并分析原因。

(4)创建一个班里的学生互助表，规定：包括学生编号，学生姓名，学生的帮助对象，每个学生有且仅有一个帮助对象，帮助对象也必须是班里的学生。（表的自参照问题）

(5)学校学生会的每个部门都有一个部长，每个部长领导多个部员，每个部只有一个部员有评测部长的权利，请给出体现这两种关系（领导和评测）的两张互参照的表的定义。（两个表互相参照的问题）

2.实验过程

1. 用alter table语句将SC表中的on delete cascade改为on delete no action,重新插入SC的数据（按照实验一）。再删除Stu_Union中sno为'10001'的数据。观察结果，并分析原因。

- 按照实验一创建 Stu_Union 表和 Course 表：

```

USE School;

CREATE TABLE Stu_Union(
    sno CHAR(5) NOT NULL UNIQUE,
    sname CHAR(8),
    ssex CHAR(1),
    sage INT,
    sdept CHAR(20),
    CONSTRAINT PK_Stu_Union PRIMARY KEY(sno)
);

INSERT INTO Stu_Union VALUES
('10001', '李勇', '0', 24, 'EE'),
('95002', '主敏', '1', 23, 'CS'),
('95003', '主浩', '0', 25, 'EE'),
('95005', '王杰', '0', 25, 'EE'),
('95009', '李勇', '0', 25, 'EE');

SELECT * FROM Stu_Union;

```

100 %

结果 消息

	sno	sname	ssex	sage	sdept
1	10001	李勇	0	24	EE
2	95002	主敏	1	23	CS
3	95003	主浩	0	25	EE
4	95005	王杰	0	25	EE
5	95009	李勇	0	25	EE

```

USE SCHOOL;

CREATE TABLE Course (
    cno CHAR(4) NOT NULL UNIQUE,
    cname VARCHAR(50) NOT NULL,
    cpoints INT,
    CONSTRAINT PK PRIMARY KEY (cno)
);

INSERT INTO Course VALUES ('0001', 'ComputerNetworks', 2);
INSERT INTO Course VALUES ('0002', 'Database', 3);

Select * from Course

```

100 %

结果 消息

	cno	cname	cpoints
1	0001	ComputerNetworks	2
2	0002	Database	3

- 按照实验1, 创建表 SC

```

use School
CREATE TABLE SC(
    sno CHAR(5) references Stu_Union(sno) on delete cascade,
    cno CHAR(4) references Course(cno) on delete cascade,
    grade INT,
    CONSTRAINT PK_SC PRIMARY KEY (sno, cno)
);

```

命令已成功完成。

完成时间: 2024-11-04T16:51:05.7263138+08:00

- alter table 语句更改外键约束为 on delete no action: 先找到on delete cascade外键名字, 找到后删除该外键, 重新添加一个 on delete no action 外键约束

键

PK_SC
FK_SC_cno_6D0D32F4
FK_SC_sno_6C190EBB

```
Alter table SC
drop constraint FK__SC__cno__6D0D32F4;
Alter table SC
add constraint FK_SC_cno foreign key (cno)
references Course(cno) on delete no action;
Alter table SC
drop constraint FK__SC__sno__6C190EBB
Alter table SC
add constraint FK_SC_sno foreign key (sno)
references Stu_Union(sno) on delete no action;
```

100 %

消息

命令已成功完成。

完成时间: 2024-11-04T16:58:27.4689358+08:00

- 按照实验1, 重新插入SC数据:

```
USE School
INSERT INTO SC VALUES ('95002', '0001', 2);
INSERT INTO SC VALUES ('95002', '0002', 2);
INSERT INTO SC VALUES ('10001', '0001', 2);
INSERT INTO SC VALUES ('10001', '0002', 2);
SELECT * FROM SC;
```

100 %

结果 消息

	sno	cno	grade
1	10001	0001	2
2	10001	0002	2
3	95002	0001	2
4	95002	0002	2

- 删除Stu_Union中sno为'10001'的数据: 由于采用on delete no action, 当从表中有匹配的记录时, 主表中相应的候选键不允许update/delete操作, 从表SC中有10001的记录, 不允许主表Stu_Union对含有10001的记录更新删除

```
delete from Stu_Union where sno='10001'
```

100 %

消息

消息 547, 级别 16, 状态 0, 第 1 行
DELETE 语句与 REFERENCE 约束"FK_SC_sno"冲突。该冲突发生于数据库"School", 表"dbo.SC", column 'sno'。
语句已终止。

完成时间: 2024-11-04T17:04:19.2444947+08:00

- 用alter table语句将SC表中的on delete no action改为on delete set NULL,重新插入SC的数据 (按照实验一) 。再删除Stu_Union中sno为'10001'的数据。观察结果, 并分析原因。

- 用alter table语句将SC表中的on delete no action改为on delete set NULL: 删除对应外键, 创建新的约束为on delete set NULL的外键。修改失败, 由于cno和Course的主键, 不能被设成 NULL, 则 on delete set null 失败

```
use School
Alter table SC
drop constraint FK_SC_cno;
Alter table SC
add constraint FK_SC_cno foreign key (cno)
references Course(cno) on delete set NULL;
Alter table SC
drop constraint FK_SC_sno
Alter table SC
add constraint FK_SC_sno foreign key (sno)
references Stu_Union(sno) on delete set NULL;
```

100 %

消息

消息 1761, 级别 16, 状态 0, 第 4 行
由于一个或多个引用列不可为 Null, 因此无法使用 SET NULL 引用操作创建外键 "FK_SC_cno"。
消息 1750, 级别 16, 状态 1, 第 4 行
无法创建约束或索引。请参阅前面的错误。

完成时间: 2024-11-09T09:51:26.2287984+08:00

3. 建立事务T3, 修改ICBC_Card表的外键属性, 使其变为on delete set NULL,尝试删除students表中一条记录。观察结果, 并分析原因。

- 按照实验2, 创建 Stu_Card 表, 创建 ICBC_Card表, 修改 choices 的原有约束

```
USE school;

CREATE TABLE Stu_Card (
card_id CHAR(14),
stu_id CHAR(10) REFERENCES students(sid) ON DELETE CASCADE,
remained_money DECIMAL(10, 2),
CONSTRAINT PK_stu_card PRIMARY KEY (card_id)
);

INSERT INTO Stu_Card VALUES ('05212567', '800001216', 100.25);
INSERT INTO Stu_Card VALUES ('05212222', '800005753', 200.50);

SELECT * FROM Stu_Card;
```

100 %

结果 消息

	card_id	stu_id	remained_money
1	05212222	800005753	200.50
2	05212567	800001216	100.25

```
USE school;

CREATE TABLE ICBC_Card (
bank_id CHAR(20),
stu_card_id CHAR(14) REFERENCES Stu_Card(card_id) ON DELETE CASCADE,
restored_money DECIMAL(10, 2),
CONSTRAINT PK_Icbc_card PRIMARY KEY (bank_id)
);

INSERT INTO ICBC_Card VALUES ('9558844022312', '05212567', 15000.1);
INSERT INTO ICBC_Card VALUES ('9558844023645', '05212222', 50000.3);
SELECT * FROM ICBC_Card;
```

100 %

结果 消息

	bank_id	stu_card_id	restored_money
1	9558844022312	05212567	15000.10
2	9558844023645	05212222	50000.30

```
alter table CHOICES drop FK_CHOICES_STUDENTS;
alter table CHOICES add constraint FK_CHOICES_STUDENTS
FOREIGN KEY(sid) references STUDENTS(sid) on delete cascade;
```

100 %

消息

命令已成功完成。

完成时间: 2024-11-04T17:28:17.3528922+08:00

- 创建事务T3：修改ICBC_Card表的外键属性，使其变为on delete set NULL，尝试删除students表中一条记录

```
Begin Transaction T3
alter table ICBC_Card drop FK_ICBC_Card_stu_c_75A278F5;
alter table ICBC_Card add constraint FK_ICBC_Card_stu_c
FOREIGN KEY(stu_card_id) references Stu_Card(card_id) on delete set NULL;
DELETE from STUDENTS where sid='800005753';
Commit Transaction T3
```

(1 行受影响)

完成时间：2024-11-04T17:34:07.9847138+08:00

- 观察结果：

- 由于Stu_card的外键约束是级联删除，当删除主表记录时，相关从表记录也会被删除，在students删除sid=800005753时，对应的从表记录（Stu_card=0512222）被删除

```
select * from Stu_Card
```

	card_id	stu_id	remained_money
1	05212567	800001216	100.25

- 由于ICBC_Card参照Stu_Card的属性card_id的外键约束是On delete set NULL，则当card_id=0512222的记录在Stu_card被删除时，从表ICBC_Card中对应记录的card_id被置为NULL

```
select * from ICBC_Card
```

	bank_id	stu_card_id	restored_money
.	9558844022312	05212567	15000.10
!	9558844023645	NULL	50000.30

4. 创建一个班里的学生互助表，规定：包括学生编号，学生姓名，学生的帮助对象，每个学生有且仅有一个帮助对象，帮助对象也必须是班里的学生。（表的自参照问题）

- 每个学生有且仅有一个帮助对象，帮助对象也必须是班里的学生：指明可以创建表STUDEN_HELP包含属性stu_id,stu_help_id。stu_help_id要参照关系STUDEN_HELP的stu_id，表示参照对象也要是班里的学生（假设全班同学都参与互助，互助表中包含全班同学的id）
- 先创建没有参照的表，注意指定stu_id(学生编号)为主键，不然无法为表添加外键参照stu_id

```
USE School
create table STUDEN_HELP(
stu_id char(20),
stu_name char(20),
stu_help_id char(20),
constraint PK_STU_ID primary key(stu_id)
)
Alter table STUDEN_HELP
add constraint FK_SC_stu_help_id_stu_id foreign key (stu_help_id)
references STUDEN_HELP(stu_id) on delete set NULL;
```

消息 1785，级别 16，状态 0，第 10 行
将 FOREIGN KEY 约束 'FK_SC_stu_help_id_stu_id' 引入表 'STUDEN_HELP' 可能导致循环或多重级联路径。请指定 ON DELETE NO ACTION 或 ON UPDATE NO ACTION，或修改其他 FOREIGN KEY 约束。
消息 1750，级别 16，状态 1，第 10 行
无法创建约束或索引。请参阅前面的错误。

- 出现提醒错误，不能将其设为on delete set NULL，要改成on delete no action

```
USE School
Alter table STUDEN_HELP
    add constraint FK_SC_stu_help_id__stu_id foreign key (stu_help_id)
    references STUDEN_HELP(stu_id) on delete no action
```

100 %

消息

命令已成功完成。

完成时间: 2024-11-04T17:58:38.9363904+08:00

5. 学校学生会的每个部门都有一个部长，每个部长领导多个部员，每个部只有一个部员有评测部长的权利，请给出体现这两种关系（领导和评测）的两张互参照的表的定义。（两个表互相参照的问题）

- 前提：没有兼任的学生会员
- 领导表：部长的工号，所属部门，部长领导的部员的工号
 - 主键为部长的工号，所属部门，部长领导的部员的工号
 - 外键参考评测表的部长的工号（部长是唯一的，不同部门不同部长，所有的部长都会被测评，则评测表中含有所有部长取值）
- 评测表：部长的工号，部门（唯一的限制），组员。保证一个部门只有一个部员有评测部长的权利，一个部门在评测表中只有一个元组
 - 主键为部长工号，可以区分一个评测关系
 - 外键参照领导表的部长的工号，所属部门，部员的工号（只有一个评测表记录在领导表中存在，也即领导关系先成立，被部长领导的部员才能评测部长）
- 先创建不带外键参照的领导表，再创建有外键参照的评测表，最后为领导表加上参照，拼写报错导致alter table..命令失败，重新执行

```
USE School
create table leadership(
    leader_id char(20),
    department char(20),
    member_id char(20),
    constraint PK_LEADERSHIP Primary key(leader_id,department,member_id)
);
create table EvaluatedBy(
    leader_id char(20),
    department char(20) UNIQUE,
    member_id char(20) ,
    constraint PK_EvaluatedBy Primary key(leader_id),
    constraint FK_evalu_leader foreign key(leader_id,department,member_id) references leadership(leader_id,department,member_id)
);
alter table leadership add constrain FK_leader_evalu
foreign key(leader_id) references EvaluatedBy(leader_id)
```

100 %

消息

消息 2715, 级别 16, 状态 6, 第 15 行
第 4 个列、参数或变量: 找不到数据类型 FK_leader_evalu.

完成时间: 2024-11-04T20:04:21.8127971+08:00

```
)
alter table leadership add constraint FK_leader_evalu
foreign key(leader_id) references EvaluatedBy(leader_id)
```

100 %

消息

命令已成功完成。

完成时间: 2024-11-04T20:08:05.6192747+08:00

3.实验总结

- 参照关系：关系不允许引用不存在的实体，在两个相关联的表之间创建约束，确保从表中每条记录的外键值为外表中存在的。外键在外表中是主键

- REFERENCES 关系R2(参照属性a2) on delete cascade
- #R2为主表

2. `on delete cascade`: 设置级联删除, 当主表中的一条记录被删除时, 如果设置了级联删除, 那么所有从表中引用该主表记录的外键记录也会被自动删除。
3. `on delete no action`: 从表中有匹配记录时, 主表中相应候选键不允许更新删除操作
4. `on delete set null`: 当主表中的一条记录被删除时, 那么所有从表中引用该主表记录的外键记录也会被自动设成NULL
5. 删除和添加外键:

```
1 Alter table SC
2     drop constraint FK_SC_cno;
3 Alter table SC
4     add constraint FK_SC_cno foreign key (cno)
5     references Course(cno) on delete set NULL;
```

6. 相互参照的表: 在创建表的时候直接两个表相互参照会报错, 创建表的时候其中一个表先不建立参照关系, 两个表都建好后, 再给那个表添加参照关系