



## 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

## 一：实验要求

---

- 1.在school数据库中建立一张新表class，包括class\_id(varchar(4)), name(varchar(10)), department(varchar(20))三个列，并约束class\_id为主键。
- 2.创建事务T3，在事务中插入一个元组 ('0001', '01CSC','CS') ,并在T3中嵌套创建事务T4，T4也插入和T3一样的元组，编写代码测试，查看结果。
- 3.在表class中，尝试设置name='01CSC'的记录的class\_id 为NULL，查看结果
- 4.在表class中，不创建事务，插入两个元组 ('0002', '01CSC'. 'CS') , ('0002', '03CSC', 'CS') , 然后查看表中几条记录，为什么？
- 5.在表class中，创建事务，并设置开启回滚，然后插入两个元组 ('0003', '03CSC'. 'CS') , ('0001', '03CSC', 'CS') , 查看结果，表中有几条记录？
- 6.在完成上面几步的前提下，尝试设置'name'为主键，看能否成功，并思考原因。

## 二：实验过程

---

1. .在school数据库中建立一张新表class，包括class\_id(varchar(4)), name(varchar(10)), department(varchar(20))三个列，并约束class\_id为主键
  - 成功创建表class，并且为 class\_id 列创建一个名为 classid 的主键约束。

```
Use School
create table class(
  class_id varchar(4),
  name varchar(10),
  department varchar(20),
  constraint classid primary key(class_id)
);
```

100 %

消息

命令已成功完成。

完成时间: 2024-10-28T16:43:39.2326800+08:00

dbo.class
列
class_id (PK, varchar(4), not null)
name (varchar(10), null)
department (varchar(20), null)
键
classid

2. 创建事务T3, 在事务中插入一个元组 ('0001', '01CSC','CS') ,并在T3中嵌套创建事务T4, T4也插入和T3一样的元组, 编写代码测试, 查看结果

- (两次事务都开启了回滚) 插入失败, 因为主键的唯一属性不允许插入两个主键 class\_id 都一样的元组

```
use school
SET XACT_ABORT ON
BEGIN TRANSACTION T3
insert into class values ('0001', '01CSC', 'CS')
SET XACT_ABORT ON
BEGIN TRANSACTION T4
insert into class values ('0001', '01CSC', 'CS')
COMMIT TRANSACTION T4
COMMIT TRANSACTION T3
```

100 %

消息

(1 行受影响)  
消息 2627, 级别 14, 状态 1, 第 7 行  
违反了 PRIMARY KEY 约束"classid"。不能在对象"dbo.class"中插入重复键。重复键值为 (0001)。

完成时间: 2024-10-28T16:49:45.6813524+08:00

两个事务都失败回滚, 没有任何元组插入 class

```
use School
select *
from class
```

100 %

结果 消息

class_id	name	department
----------	------	------------

3. 在表class中, 尝试设置name='01CSC'的记录的class\_id 为NULL, 查看结果

- 先插入一个name='01CSC'的元组, 再尝试将其的class\_id改为NULL

- 结果：由于class\_id为主键，主键的NOT NULL约束导致更改失败
- （执行完后将该元组从class中删除）

```
use School
insert into class values ('0001','01CSC','CS')
UPDATE class SET class_id=NULL where name='01CSC'
```

100 %

消息

(1 行受影响)  
 消息 515, 级别 16, 状态 2, 第 3 行  
 不能将值 NULL 插入列 'class\_id', 表 'School.dbo.class'; 列不允许有 Null 值。UPDATE 失败。  
 语句已终止。

完成时间: 2024-10-28T16:55:15.8365317+08:00

4. 在表class中，不创建事务，插入两个元组（'0002'，'01CSC'，'CS'），  
 （'0002'，'03CSC'，'CS'），然后查看表中有几条记录，为什么？

- 向class中插入两个元组，由于两次插入的主键class\_id相同，违反了主键的唯一性，只有第一次插入成功，第二次插入失败

```
use School
insert into class values ('0002','01CSC','CS')
insert into class values ('0002','03CSC','CS')
```

100 %

消息

(1 行受影响)  
 消息 2627, 级别 14, 状态 1, 第 3 行  
 违反了 PRIMARY KEY 约束"classid"。不能在对象"dbo.class"中插入重复键。重复键值为 (0002)。  
 语句已终止。

完成时间: 2024-10-28T17:00:22.9956331+08:00

- 由于两次插入不是事务，第二次插入失败不用回滚，则第一次插入成功，表中只有插入的第一次的元组

```
use School
select *
from class
```

100 %

结果 消息

	class_id	name	department
1	0002	01CSC	CS

5. 在表class中，创建事务，并设置开启回滚，然后插入两个元组（'0003'，'03CSC'，'CS'），  
 （'0001'，'03CSC'，'CS'），查看结果，表中有几条记录？

- 注意：第2个实验任务由于事务中插入元组主键相同，所以没有插入class\_id=0001的元组，第3个实验任务中也将class\_id=0001的元组在实验后删去
- 事务执行成功

```

use school
SET XACT_ABORT ON
BEGIN TRANSACTION T5
insert into class values ('0003', '03CSC', 'CS')
insert into class values ('0001', '03CSC', 'CS')
COMMIT TRANSACTION T5

```

100 %

消息

(1 行受影响)

(1 行受影响)

完成时间: 2024-10-28T17:06:24.2118175+08:00

- class中有三条记录，本任务插入的两个元组在class中

```

use school
select *
from class

```

100 %

结果 消息

	class_id	name	department
1	0001	03CSC	CS
2	0002	01CSC	CS
3	0003	03CSC	CS

6. 在完成上面几步的前提下，尝试设置'name'为主键，看能否成功，并思考原因。

- 为了设置name为主键，要删掉原有的classid的主键
- 设置失败：因为一开始创建表的时候没有设置name列的not null约束，导致name列可以设为null，这个违反了主键的不为null的属性，在当前的环境下无法添加name为主键

```

dbo.class
列
class_id (varchar(4), not null)
name (varchar(10), null)
department (varchar(20), null)

```

```

use school
Alter table class drop
constraint classid
Alter table class add
constraint classname primary key([name])

```

100 %

消息

消息 8111, 级别 16, 状态 1, 第 2 行  
无法在表 'class' 中可为 Null 的列上定义 PRIMARY KEY 约束。

消息 1750, 级别 16, 状态 0, 第 2 行  
无法创建约束或索引。请参阅前面的错误。

完成时间: 2024-10-28T17:17:39.4649955+08:00

### 三：实验总结

- NULL 不等于空格和0
- 违反主键的唯一性属性和 NOT NULL 属性，会破坏实体完整性
- 事务建立：

```
1 | BEGIN TRANSACTION T1
2 | ....
3 | COMMIT TRANSACTION T1
```

事务回退机制：为ON时，事务有语句错误，整个事务终止回滚，为OFF，只回滚产生错误的SQL语句

```
1 | SET XACT ABORT ON/OFF
```

- 当与现有数据环境不等时，无法建立实体完整性以及参照完整性