

数据库实验lab8

曹广杰

15352015 数据科学与计算机

授课教师：刘玉葆

Content

数据库实验lab8

Content

实验目的

句法汇总

实验内容

为table设置primary key

事务与实体完整性

外键的建立

实验目的

1. 学习实体完整性的建立
2. 了解实践违反实体完整性的结果

句法汇总

1. 为table设置主键以及设置主键的要求。
2. varchar与char的不同。
3. 设置主键的语法格式。
4. 创建事务与设置回滚选项。
5. 外键的建立条件。

实验内容

为table设置primary key

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))

约束class_id为主键

```

1 create table class
2 (
3     class_id varchar(4),-- not null unique,
4     name varchar(10),
5     department varchar(20)
6     constraint hahaha primary key (class_id)
7 )

```

这里使用的就是常规的创建表单的语法。

注:

1. 选择class_id为主键

选择class_id为主键，规范的写法需要在class_id的声明处添加限定：`not null unique`；

但是不加也可以，在sql系统中，一旦该属性设置为主键了，那么系统会替我们完成这一项工作。

2. varchar，更加灵活的字符。

该数据结构会将传入的小于当前变量字符串长度的参数填满，使之与当前的字符串长度一致。

3. 设置主键:

句法: `constraint` `自定义的表示主键关系的字符串` `primary key` (`属性`)

事务与实体完整性

创建事务T3

- 在T3中插入一个元组 (`'00001'`, `'01CSC'`, `'CS'`);
- 在T3中嵌套创建事务T4
 - 在T4插入和T3一样的元组

```

1 set xact_abort on
2 begin transaction t3
3     insert into class
4     values ('0001', '01csc', 'CS')
5     begin transaction t4
6         insert class
7         values ('0001', '01csc', 'CS')

```

运行结果与分析:

该事务不能正常运行，原因，插入了两个完全一样的元组，这违反了实体完整性——即如果插入成功，则该table中的主键就不具有区分度，不能保证unique的特性，笔者实验中修改了class_id为任何非'0001'的数据之后都是可以插入的。按师姐的话说，那违反了数据的对应性。

句法:

1. set `xact_abort` on

`xact_abort` 作为一个设置选项。

在打开之后，如果事务运行不成功，则终止事务的运行并回滚到事务运行之前；

若关闭，即on改为off，则事务不再同生死共患难，某条语句出错，只回滚到出错语句之前。

2. transaction

transaction，事务，交易。这里是指一系列的用于增、删、改当前数据库的命令语句，事务的出现可以使修改语句不再分散独立。配合xact_abort的设置，事务可以简化出错之后的恢复。

3. insert与insert into

没什么区别，笔者亲试。

查询的结果是每一个学号对应一个c++课程的成绩，这里Null是存在的，在升序排列中，Null在最上面，被系统当做最小的值处理。

外键的建立

类似于前文所述，外键的使用也有要求，由于该表的外键是另一个表的主键，显然外键也会满足“not null unique”的特性。如果当前table中存在另一个表主键所不包含的数据时，是不可以建立外键连接的。